

Konstantin Hahl

PAIKANNUSSATELLIITTEN RATAENNUSTUKSET KONEOPPIMISEN AVULLA

Tekniikan ja luonnontieteiden tiedekunta
Diplomityö
Toukokuu 2019

TIIVISTELMÄ

Konstantin Hahl: Paikannussatelliittien rataennustukset koneoppimisen avulla
Diplomityö
Tampereen yliopisto
Teknis-luonnontieteellinen DI-tutkinto-ohjelma
Toukokuu 2019

Satelliittipaikannuksen välttämättömyys on, että paikannussatelliittien sijainnit tunnetaan paikannushetkenä. Tätä varten satelliitit lähettävät tietoa radoistaan käyttäjille kuuden sekunnin mittaisina viesteinä 30 sekunnin intervallilla. Paikannuksen aloittamiseksi käyttäjän on siis odotettava ratatietojen vastaanottamista. Tätä odotusaikaa voidaan lyhentää lataamalla satelliittien ratatiedot internetin välityksellä, mutta kaikilla paikannuslaitteilla ei ole tähän mahdollisuutta, kuten tavallisella auton navigaattorilla tai urheilukellolla. Toinen tapa lyhentää odotusaikaa on ennustaa paikannussatelliittien radat laitteella. Kun laite aloittaa paikannuksen, pystytään satelliittien sijainnit selvittämään edellisillä paikannuskerroilla tehtyjen rataennustusten avulla. Rataennustusten tarkkuus luonnollisesti huononee ennustusajan pidentyessä, mutta riittävään tarkkuuteen päästään vielä muutaman viikon mittaisilla ennustuksilla.

Tässä diplomityössä tutkitaan, miten koneoppimista voidaan hyödyntää paikannussatelliittien ratojen ennustuksessa, jota tehdään paikannuslaitteella. Tarkoituksena on tuottaa paikannuslaitteille avusteparametreja, joilla pystytään parantamaan laitteella tehtäviä satelliittien rataennustuksia. Parametrien tulisi olla tiedostokoiltaan tarpeeksi pienikokoisia sekä pitkäikäisiä, jotta niiden lähettäminen laitteelle olisi kannattavaa. Ehtona on myös, että parametrien lisäksi laite hyödyntää vain reaaliaikaista lähetysefemeridiä (*broadcast ephemeris*), eli internetyhteyttä parametrien vastaanottamisen jälkeen ei ole mahdollisuutta käyttää eikä tallettaa laitteeseen aikaisemmin vastaanotettuja efemeridejä.

Ongelmaa on lähestytty siten, että pyritään löytämään koneoppimisen avulla käytössä olevasta ennustusmallista mallinnettavia virheitä. Rataennustuksen virheestä tehdään korjausmalli, jonka funktiona on vain lähetysefemerideistä saatuja parametreja. Laitteella tehdystä ennustuksesta voidaan näin ollen poistaa korjausmallin ennustama virhe, jolloin saadaan parempi ennustus laitteelle. Tarkoitus on selvittää, miten malli suoriutuu 14 vuorokauden mittaisissa ennustuksissa.

Työssä havaittiin, että käytetyjen rataennustusten virhe on selvästi aikariippuvainen. Korjausmallin syötteenä on ainoastaan satelliitin tunnistenumero sekä aika kolmessa eri muodossa: vuodenaika, vuorokaudenaika sekä GPS-sekunti. Mallin ulostulon parametrit on pyritty tasapainottamaan ennustustarkkuuden ja parametrien määrän välillä.

Koneoppimisen metodina on käytetty satunnaista metsää (*random forest*), joka on nykystandardeilla täysin varteenotettava vaihtoehto sovelluksissa neuroverkkojen ohella. Sen etuna neuroverkkoihin on huomattavasti nopeampi opetus aika sekä se ettei se ole neuroverkkojen tapaan "musta laatikko", vaan korjausmallin rakennetta voidaan tarkastella yksityiskohtia myöten.

Mallilla saadaan parannettua ennustustarkkuutta huomattavasti. Jokainen virhekomponentti (radiaali-, tangentiali- ja normaalivirhe) saadaan noin puolitetuksi 14 vuorokauden ennustuksissa, sekä parametreja voidaan tuottaa laitteelle tarpeeksi ajoissa tulevaisuuden ennustuksia varten.

Avainsanat: GNSS, GPS, satelliitin rataennustus, koneoppiminen, random forest

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

ABSTRACT

Konstantin Hahl: Orbit prediction of positioning satellites using machine learning
Master of Science Thesis
Tampere University
Science and Engineering, MSc
May 2019

The necessity of the satellite positioning is that the locations of the positioning satellites are known at the time of positioning. For this purpose, satellites transmit information about their orbits to the users with six-second long messages every 30 second. Thus, in order to start positioning, the user must wait for the data to be received. This waiting time can be shortened by downloading the satellite orbit information via the Internet, but not all of the positioning devices have this possibility, for example a regular car navigator or a sports watch. Another way to shorten the waiting time is to predict the orbits of the satellites on the device. When the device starts the positioning, it is possible to locate the satellites with the help of orbit predictions made during the last positioning session. The prediction accuracy naturally decrease when the prediction length increases, but sufficient accuracy can be achieved with a few weeks long predictions.

In this master's thesis, we are studying how we can use machine learning in the orbit prediction of GNSS-satellites. The aim is to provide assistive parameters to improve orbit prediction capabilities on the device. The file size of the parameters should be small enough and the parameters themselves long lasting so that sending them to the device would be worth it. It is also assumed that, in addition to the parameters, we only use real-time *broadcast ephemeris*, so we do not have the possibility to access the internet after the parameters have been received. Also the device itself cannot store previously received broadcast ephemerises.

The problem has been approached by trying to find a model to predict errors in the current orbit prediction model through machine learning. The goal is to make a model that predicts our prediction error using only information in real-time broadcast ephemeris as an input. Therefore, a prediction made on a device can be improved using the predicted error by the model to get a better prediction to the device. In this thesis, we examine how the model performs in predictions of 14 days.

It was found in this master's thesis that the error of our orbit prediction is clearly time-dependent. Our model's input is only the identifying number of the satellite, and time in three different formats: time of year, time of day and GPS second. The output parameters of the model have been chosen to be the best compromise between the prediction accuracy and the size of the parameters.

The method we used was *random forest*, which is a fully-fledged alternative to applications in addition to neural networks. Its advantage compared to neural networks is considerably faster training time, and not being a "black box" like neural networks, so we can look at the structure of the model.

Our model performs well. Each error component (radial, tangential and normal error) is approximately halved in predictions of 14 days. We are also able to create parameters long before they are used.

Keywords: GNSS, GPS, satellite orbit prediction, machine learning, random forest

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Diplomityö on aloitettu Tampereen teknillisen yliopiston matematiikan laboratorion alaisuudessa. Työn loppuvaiheilla Tampereen korkeakoulut yhdistyivät samaksi kokonaisuudeksi, Tampereen yliopistoksi. Työskentelin paikannusalgoritmien tutkimusryhmässä vuosina 2017-2019, jonka loppuvaiheilla tämä diplomityö luotiin.

Työn ohjaajina toimivat professori Robert Piché ja yliopistolehtori Simo Ali-Löytty. Kiitän heitä panoksestaan tähän työhön. Kiitos myös paikannusalgoritmien tutkimusryhmän henkilöstölle, jotka ovat konsultoineet minua työn eri vaiheissa.

Työ on tehty yhteistyössä yrityksen HERE Technologies kanssa, joka on rahoittanut tutkimukseni.

Tampereella, 3. toukokuuta 2019

Konstantin Hahl

SISÄLLYSLUETTELO

Lyhenteet ja merkinnät	v
1 Johdanto	1
2 Teoria	3
2.1 Kernel	3
2.2 Oppimispuu	4
2.2.1 Perusteet	4
2.2.2 Puun luominen	5
2.2.3 Satunnainen metsä	6
3 Tutkimusongelma ja sen tausta	9
3.1 Satelliittipaikannuksen periaate	9
3.2 Tutkittava ongelma	13
3.3 Aiemmat tutkimukset	19
4 Mallin luominen	21
4.1 Parametrien valinta	21
4.2 Mallin ulostulo	23
4.2.1 Tangentiaalivirhe	24
4.2.2 Radiaali- ja normaalivirhe	25
4.3 Datan esikäsittely	27
5 Mallin suorituskky	30
5.1 Parannus virhekomponenteittain	31
5.2 SISRE	37
5.3 Vertailu konvoluutioverkolla tuotettuun korjausmalliin	41
5.4 Parametrien luonti etukäteen	42
5.5 Opetusdatan koko	43
6 Yhteenveto	47
Lähdeluettelo	48

LYHENTEET JA MERKINNÄT

$ \cdot $	joukon alkioden lukumäärä
$\ \cdot\ $	vektorin normi
$\underline{0}$	ollevektori
$\underline{\alpha}, \underline{\beta}$	reaaliarvoinen vakiovektori
b	kellosta johtuvan etäisyysmittavirhe
b_0	kellosta johtuvan etäisyysmittavirheen estimaatti
BE	<i>Broadcast ephemeris</i> , satelliittin lähettämä tieto radastaan, kellostaan yms.
CART cost	<i>Classification And Regression Trees</i> , luokittelu- ja regressiopuut kustannusfunktio
\mathcal{D}	opetusdatajoukko
d	vakio
$\mathcal{D}_{L/R}$	joukon vasemman- tai oikeanpuolinen osajoukko solmun jaossa
D	opetusdatan ominaisuuksien lukumäärä
Δ	hyötyfunktio
det	determinantti
ϵ	virhe
f	funktio
G	geometriamatriisi
GLONASS	<i>Globalnaja Navigatsionnaja Sputnikovaja Sistema</i> , venäläinen globaali satelliittipaikannusjärjestelmä
GNSS	<i>Global Navigation Satellite System</i> , globaali satelliittipaikannusjärjestelmä
GPS	<i>Global Positioning System</i> , yhdysvaltalainen globaali satelliittipaikannusjärjestelmä
Imp	hyödyllisyysfunktio
IoT	<i>Internet of Things</i> , asioiden internet
j	ominaisuus
k	kernel-funktio
μ	tunnettu vertailumuuttuja kernel-funktiossa
ΔN	satelliitin paikkaestimaatin normaalivirhe
ρ	pseudoetäisyys
ϕ	kernel-verkko
PE	<i>Precise Ephemeris</i> , myöhemmin lasketut tarkemmat tiedot satelliittien radoista
PHIP	<i>Piecewise Cubic Hermite Interpolating Polynomial</i> , toisen asteen sovitus
PRN	<i>Pseudorandom Noise</i> , uniikki signaalin koodaus, käytetään myös satelliittien numerointiin
rank	matriisin aste
ΔR	satelliitin paikkaestimaatin radiaalivirhe

σ	keskihajonta
SISRE	<i>Signal-In-Space Range Error</i> , satelliitin ratavirheestä aiheutuva virhe käyttäjän paikantamisessa
SRP	<i>Solar Radiation Pressure</i> , auringon säteilypain
\mathcal{T}	mahdollisten kynnysarvojen joukko
T	puiden joukko satunnaisessa metsässä
t	kynnysarvo, satunnaisen metsän puu
ΔT	satelliitin paikkaestimaatin tangentiaalivirhe
\mathbf{v}_m	solmujen leikkausten muuttujat ja kynnysarvot puun latvasta lehteen
w_m	arvoalueen m keskiarvo
w_R	radiaalivirheen kerroin SISRE:ä laskettaessa
$w_{T,N}$	tangentiaali- ja normaalivirheen kerroin SISRE:ä laskettaessa
WLAN	<i>Wireless Local Area Network</i> , langaton lähiverkko
X	vekrotiavaruus
$\mathbf{x}, \mathbf{y}, \dots$	vektori
\mathbf{x}_0	käyttäjän paikkaestimaatti
$\delta \mathbf{x}$	käyttäjän paikkaestimaatin muutos
x_i	vektorin \mathbf{x} i :nnes alkio
\mathbf{x}_j	opetusdatan j :nnes syötevektori
\bar{y}	muuttujan y keskiarvo

1 JOHDANTO

Satelliittipaikannuksesta on tullut tärkeä osa nykypäiväistä elämää ja sen merkitys arjessa kasvaa päivä päivältä. Sitä käytetään paikantamaan ajoneuvoja, tavaroita, ihmisiä ja luontoa. Ajoneuvojen ja kännyköiden lisäksi myös IoT-laitteet (*Internet of Things*) käyttävät satelliittipaikannusta hyödyksi. Maan kiertoradalla on jo yli 100 paikannussatelliittia useammasta eri satelliittikonstellaatiosta, ja lisää satelliitteja laukaistaan aika ajoin kiertoradoille. Satelliittikonstellaatiolla tarkoitetaan ryhmää satelliitteja, jotka ovat saman tahon hallinnassa ja operoitu siten, että toimisivat yhdessä mahdollisimman hyvin käyttäjän paikannusta varten. Tunnetuin satelliittikonstellaatio on Yhdysvaltain puolustusministeriön ylläpitämä GPS (*Global Positioning System*). [18]

Vastaanottimen kolmiulotteisen paikan laskemiseen tarvitaan mittaus vähintään neljän satelliitin etäisyydestä laitteeseen. Etäisyysmittauksista saadaan epälineaarinen yhtälöryhmä, jossa tuntemattomina arvoina ovat käyttäjän paikka kolmiulotteisessa avaruudessa sekä kellonaika. Satelliitit lähettävät joka kuudes sekunti kellonaikaansa, josta vastaanotin laskee satelliitin ja vastaanottimen välisen *pseudoetäisyyden* ρ , kun tiedetään radiosignaalin kulkevan noin valonnopeutta. [18] Vastaanottimen kaksiulotteinen paikka voitaisiin laskea vastaavasti, jos oletetaan laitteen olevan Maan pinnalla. Myös jos on saatu lähihistoriassa mittaushavainto jostain satelliitista, johon menetetään yhteys, oletamalla pseudoetäisyyden satelliittiin pysyvän samana, voidaan aproksimoida käyttäjän sijainti ilman neljättä satelliittia. Tällöin tietenkin tarkkuus huononee.

GPS-satelliitit lähettävät tietoa omasta radastaan ja kellonsa virhettä 30 sekunnin mittaisilla viesteillä. Tästä tietopaketista käytetään nimitystä *lähetys efemeridi* (BE, *broadcast ephemeris*). Konstellaatiosta riippuen satelliitit lähettävät samaa efemeriditietoa vartista tuntiin, ja vastaanotinlaite pystyy vastaanotettujen Keplerin rataparametrien avulla laskemaan reaaliajassa satelliitin nykyisen sijainnin. Efemeriditiedot ovat käyttökelpoisia vartista tunteihin, jonka jälkeen niillä lasketut radan virheet kasvavat liian suuriksi paikannukseen käytettäväksi. [18]

Juuri paikannuksen aloittanut laite joutuu siis odottamaan maksimissaan 30 sekuntia ennen kuin se pystyy laskemaan satelliittien paikkoja ja aloittamaan paikannusta. Huonossa tapauksessa laite ei välttämättä onnistu vastaanottamaan efemeriditietoja, jos satelliitin ja vastaanottimen välillä on esteitä, esimerkiksi rakennuksia, suuria puita, tai vastaanotin sattuu olemaan tunnelissa. Tällöin paikannuksen aloitus viivästyy 30 sekunnilla, koska laite joutuu odottamaan, että ratatiedot lähetetään uudestaan.

Internetyhteydellä laite pystyisi hakemaan lähetysefemeridin tiedot internetistä, mutta usealla navigointilaitteella tätä ominaisuutta ei ole, esimerkiksi normaalilla auton navigaattorilla. Suomessa monet puhelinliittymät tarjoavat tiedonsiirron kiinteään hintaan, mutta isossa osaa maailmaa näin ei vielä ole, vaan käyttäjä maksaa tiedonsiirrosta kulutuksen mukaan. Tällöin siis ei internetyhteyskään ole aukoton ratkaisu ongelmaan.

Nykyinen ratkaisu on vastaanotinlaitteessa tapahtuva satelliittien ratojen ennustus. Fysiikaalisten voimamallien avulla satelliitin rataa pystytään ennustamaan tarkasti muutaman viikon edestä, jonka jälkeen virhe kasvaa paikannustarkoitukseen liian suureksi. Voimamallissa on otettu huomioon isoimmat tekijät: Maan, Auringon, Kuun vetovoimat ja Auringon säteilypain. [20] Maata lähellä olevien planeettojen vetovoiman vaikutusta on myös tutkittu, mutta se ei tuottanut merkittävää parannusta malliin [19]. Auringon säteilypaineen mallintamisessa on tyydytty kahden parametrin malliin.

Ennustustarkkuutta pystytään parantamaan merkittävästi, kun käytössä on laitteen ulkopuolinen laskentateho ja kerätty data. Palvelimella tapahtuvalla laskennalla voidaan parantaa ennustusta estimoimalla normaalin rataennustuksen virhettä. Yksinkertainen tapa on ennustaa satelliitin paikkaa taaksepäin, jolloin voidaan verrata ennustusta todellisiin paikkoihin. Tähän virheeseen voidaan sovittaa toisen asteen käyrä, jota hyödynnetään tulevaisuuden ennustamisessa. Vastaavalla periaatteella neuroverkoille pystytään opettamaan taaksepäin ennustetulla radalla tulevaisuuteen tehtyjen ennustusten virhettä. Virhe muutetaan kolmikanavaiseksi kuvaksi, ja konvoluutiokerrosten avulla ennustetaan ennustusten virhekäyriä [16].

Nämä vaativat kuitenkin jatkuvaa datan keräämistä ja säilytystä, mikä ei vastaanotinlaitteelle ole mahdollista. Palvelimella lasketut ennustuksen voitaisiin internetin välityksellä jakaa paikannuslaitteille, mutta rajoittamaton internetyhteys ei ole kaikille paikannuslaitteille mahdollista.

Tämän diplomityön tavoitteena on parantaa laitteessa tapahtuvaa GPS-satelliittien efemeridiennustusta yhden lähetysefemeridin avulla. Tällöin itse ennustus tehtäisiin kokonaan laitteella, eikä internetyhteyttä tarvitse muuhun kuin parannusmallin parametrien päivittämiseen. Ratkaisua etsitään koneoppimisen avulla. Vanhat efemeriditiedot ovat internetissä avointa dataa, ja niiden avulla voidaan opettaa tietokonetta ennustamaan normaalin voimamallin rataennustuksen virhettä. Työn periaatetta oletettavasti pystytään soveltamaan myös muihin paikannussatelliittikonstellaatioihin.

2 TEORIA

Koneoppiminen on tekoälyn osa-alue, jossa ongelmaa ratkaistaan koneellisesti jonkin pohjatiedon avulla. Sen yksi osa-alue on *ohjattu oppiminen*, jossa tunnettujen syötteiden ja ulostulojen välille luodaan malli, joka voi olla yksinkertaisesta kompleksiseen. Nykypäivänä pystytään keräämään, varastoimaan ja käsittelemään suuria määriä dataa, minkä ansiosta koneoppimisesta on tullut kilpailukykyinen vaihtoehto mallinnusongelmien ratkaisemiseksi. [13]

Eri koneoppimisen metodeja on laajasti, eikä mikään niistä ole absoluuttisesti parempi kuin muut. Jokaisella metodilla onkin omat vahvuutensa eri haasteissa. Tässä diplomityössä tutkitaan mahdollisuutta käyttää *satunnaista metsää* ongelman ratkaisemiseen. Määrittellään opetusdatan ja mallin syötteen välisen yhteyden ydinfunktion eli kernelin avulla, josta päästään oppimispuun (*reduction tree*) ja sitä kautta satunnaisen metsän luomiseen.

2.1 Kernel

Kernel-funktio (ydinfunktio, kernel) määrittellään olevan kahden muuttujan reaaliarvoinen funktio $k(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$, missä \mathbf{x}, \mathbf{y} kuuluvat johonkin vektoriavaruuteen X . Tyypillisesti funktio on symmetrinen ($k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x})$), ja sen maalijoukko määrittellään ei-negatiiviseksi.

Joissain tapauksissa halutaan verrata muuttujan \mathbf{x} jotain ominaisuutta tunnettuun muuttajaan μ kerneleiden avulla. Voidaan siis tehdä funktio

$$f_{\mu}(\mathbf{x}) = k(\mathbf{x}, \mu). \quad (2.1)$$

Samanlaisuutta mittaavia kernel-funktioita ovat esimerkiksi

$$\begin{aligned} k(\mathbf{x}, \mathbf{y}) &= \|\mathbf{x} - \mathbf{y}\| \quad \text{ja} \\ k(\mathbf{x}, \mathbf{y}) &= \|\mathbf{x} - \mathbf{y}\|^2. \end{aligned} \quad (2.2)$$

Kernel-funktioiden avulla määrittellään kernel-verkko, joka on muotoa

$$\phi(\mathbf{x}) = [f_{\mu_1}(\mathbf{x}), \dots, f_{\mu_n}(\mathbf{x})], \quad (2.3)$$

missä siis verrataan syötemuuttujaa eri kernel-funktioilla f_{μ_i} tunnettuihin muuttujiin μ_i .

Kernel-funktioita ja -verkkoja käytetään laajasti eri koneoppimismetodeissa ja niillä pyritään löytämään syöte- ja opetusdatan väliltä riippuvuuksia. Näin voitaisiin tehdä esimerkiksi silloin, kun tutkitaan syötekuvan samankaltaisuutta opetusdatan kuviin, ja sen avulla päätellään, mitä syötekuvassa on. Tähän tarkoitukseen toimivia kernel-funktioita on kuitenkin hankalaa ja työlästä tuottaa itse, joten on suotavaa pystyä tuottamaan ne koneellisesti opetusdatan avulla. *Oppimispuu* on tähän yksi keino. [13]

2.2 Oppimispuu

Oppimispuu (reduction tree) on helposti hahmotettava koneoppimisen (ja tietorakenteen) malli, jota käytetään sekä luokittelu- että regressio-ongelmien ratkaisuun. Niiden toiminta perustuu määrittelyjoukon rekursiiviseen jaotteluun siten, että jokaiselle määrittelyjoukon osajoukolle määritellään oma lokaali vakiofunktioinen malli. Tämä esitetään puurakenteena, jossa jokainen *lehti* (*leaf*), puun päättävä haara, edustaa koko puun yhtä osajoukkoa. Puun haaraa kutsutaan *solmuksi*.

2.2.1 Perusteet

Oppimispuun käyttäminen mallina on yksinkertaista. Algoritmia käydään läpi puun latvasta alaspäin solmu kerrallaan. Kussakin solmussa verrataan, onko syötemuuttuja x_j suurempi vai pienempi kuin kynnyksarvo t_j . Vastauksesta riippuen kuljetaan saatua oksaa syvemmälle puuhun seuraaviin solmuihin. Puun solmussa on siis jaettu syötemuuttujan x_j akseli kahteen osaan, jolloin mahdollisen vastauksen osajoukko pienenee solmu kerrallaan. Oletuksena kaikki syötteet ovat reaaliarvoisia, jolloin on luontevaa verrata syötemuuttujaa x_j numeeriseen arvoon t_j . [13]

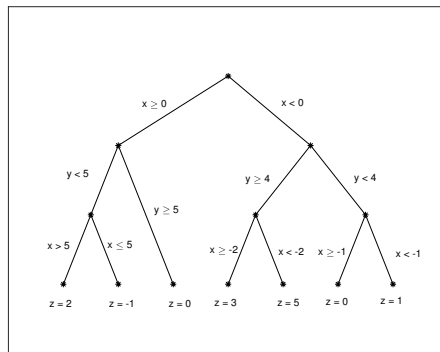
Oppimispuun toimintaa on havainnollistettu kuvassa 2.3. Ensimmäinen solmu kysyy, onko x suurempi vai pienempi kuin 0. Tästä vastauksesta riippuen seuraava kysymys on, onko y suurempi vai pienempi kuin 4 tai 5. Tämän perusteella siirrytään puussa solmu kerrallaan alaspäin, kunnes saavutetaan *lehti*, josta selviää syötteen ulostulo z . Näiden peräkkäisten akselin jakotoimenpiteiden jälkeen kaksikulotteinen avaruus saadaan jaettua kuuteen arvoalueeseen, kuten kuvasta 2.3 (b) nähdään.

Yleisessä tapauksessa kussakin arvoalueessa opetusdata ei saa yksikäsitteisiä arvoja. Voidaan kuitenkin luoda malli laskemalla kunkin arvoalueen m opetusdatan keskiarvo w_m . Tällöin malli voidaan kirjoittaa muodossa

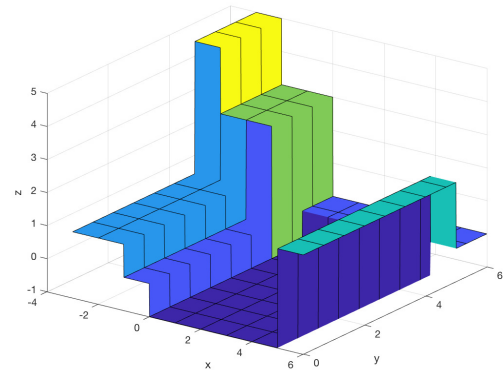
$$f(\mathbf{x}) = \sum_{m=1}^M w_m \phi(\mathbf{x}, \mathbf{v}_m), \quad (2.4)$$

missä \mathbf{v}_m viittaa solmujen leikkausten muuttujiin ja kynnyksarvoihin puun latvasta lehteen m . Kernel-verkko saa luonnollisesti arvon 1, kun muuttuja \mathbf{x} kuuluu arvoalueeseen m , ja

arvon 0 kun se ei kuulu. [13]



(a)



(b)

Kuva 2.3. Yksinkertainen esimerkki oppimispuusta. Kuvassa (a) muuttujat x ja y ovat syötteitä ja z on ulostulo. Kuvassa (b) syötteiden ja ulostulon relaatio kolmiulotteisessa avaruudessa.

Luokittelu- ja regressiopuut (*CART, Classification And Regression Trees*) ovat siis yksinkertaisia malleja, joissa yksinkertaiset funktiot määrittelevät alueen ja painokertoimet määrittelevät kunkin alueen arvon.

Tästä pystytään luomaan luokittelumalli, jossa jokainen lehti kuvaa jotakin luokkaa. Tässä diplomityössä ollaan kuitenkin enemmän kiinnostuneita regressiopuusta, joka luokan sijaan antaa ulostulona reaaliarvon.

2.2.2 Puun luominen

Oppimispuun luomisessa tarkoitus on tuottaa puu, jossa kunkin lehden ulostulo vastaisi opetusdatan arvoja kyseisessä lehdessä, mutta malli ei olisi kuitenkaan ylisovitetty opetusdataan. Luomisalgoritmi on rekursiivinen funktio, jossa puuta luodaan ylhäältä alaspäin, ja jokaisen solmun jako tehdään optimaalisesti opetusdatan \mathcal{D} avulla. Opetusdata \mathcal{D} on opetustapausten $(\mathbf{x}_{i,1:|\mathcal{D}|}, y_i)$ joukko, jossa $\mathbf{x}_{i,1:|\mathcal{D}|}$ on syöte-vektori ja y_i on ulostulo. $|\mathcal{D}|$ on opetustapausten lukumäärä.

Jakofunktio valitsee parhaan piirteet j (*feature*) ja kynnyсарvon t (*threshold*) opetusdatasta \mathcal{D} seuraavanlaisesti:

$$(j^*, t^*) = \arg \min_{j \in \{1, \dots, D\}} \min_{t \in \mathcal{T}_j} \text{cost}(\{\mathbf{x}_i, y_i : x_{ij} \leq t\}) + \text{cost}(\{\mathbf{x}_i, y_i : x_{ij} > t\}), \quad (2.5)$$

missä \mathbf{x}_i on opetusdatan yksi syötevektori, y_i tämän vaste ja D on piirteiden lukumäärä opetusdatassa \mathcal{D} . Kustannusfunktio (*cost function*) annetulle opetusdalle \mathcal{D} määritellään

$$\text{cost}(\mathcal{D}) = \sum_{i \in \mathcal{D}} (y_i - \bar{y})^2, \quad (2.6)$$

missä \bar{y} on vastemuuttujan y keskiarvo joukossa \mathcal{D} . Mahdollisten kynnysarvojen joukko \mathcal{T}_j piirteelle j on muuttujien x_{ij} joukko opetusdatassa \mathcal{D} .

Data voitaisiin myös jaotella monimutkaisemmin. Sen sijaan, että kynnysarvo on vain yksi epäyhtälö $x_{ij} > t$, voitaisiin jakaa lähtöjoukko kahden välin sijaan useampaan väliin, esimerkiksi $x_{1,j} \leq t_{j,1} < x_{2,j} \leq t_{j,2} < x_{3,j}$. Saatu puu ei olisi enää binääripuu. Tämä kuitenkin aiheuttaisi *datan pirstoutumista* (*data fragmentation*), eli data jakautuisi huomattavasti epätasaisemmin puun eri "oksastoihin", mistä voi seurata ylisovitus. [13]

Puun yksi ominaisuus on sen syvyys, eli pisin reitin pituus. Syvemmillä puilla pystytään tekemään monimutkaisempia malleja, mutta niillä on myös taipumus ylisovittaa malli opetusdataan. Puuta luodessa sille annetaan parametri, joka määrittää puun suurimman sallitun syvyyden (*max depth*). Puuntekoalgoritmit eivät kuitenkaan väkisin tee puusta mahdollisimman syvää, vaan pysäyvät "oksan" seuraavilla periaatteilla:

- Hyöty on liian pieni. Ominaisuuden käytön hyöty määritellään

$$\Delta(\mathcal{D}, s) = \text{cost}(\mathcal{D}) - \left(\frac{|\mathcal{D}_L|}{|\mathcal{D}|} \text{cost}(\mathcal{D}_L) + \frac{|\mathcal{D}_R|}{|\mathcal{D}|} \text{cost}(\mathcal{D}_R) \right), \quad (2.7)$$

missä \mathcal{D} on solmun saavuttama joukko, s on solmu, joka sisältää tiedon jaossa käytetystä muuttujasta ja kynnysarvosta. \mathcal{D}_L ja \mathcal{D}_R ovat joukon \mathcal{D} kynnysarvon vasemman- ja oikeanpuoliset osajoukot.

- Puu on saavuttanut käyttäjän määrittelemän suurimman sallitun syvyyden.
- Uuden solmun haarojen alkioden jakauma on liian samanlainen, eli solmu ei anna juurikaan uutta tietoa [12, 13].
- Uuden solmun tekemiseen tarvittava data on liian vähäistä. [13]

Optimaalinen datan jaottelu on NP-täydellinen ongelma. Datamäärät koneoppimista vaativissa ongelmissa ovat usein suuria, joten yleensä ratkaistaan vain lokaali *suurimman uskottavuuden estimointi* (*MLE, Maximum Likelihood Estimation*). Tätä keinoa käytetään myös luokittelu- ja regressiopuissa. [2, 13]

2.2.3 Satunnainen metsä

Oppimispuu toimii yksinkertaisuudessaan hyvin. Kuitenkin puun rakenne muodostuu satunnaisesti saaduista lokaaleista optimeista, jolloin mallin toimivuus vaihtelee. Oppimispuulla on myös taipumus ylisovittaa malli dataan, jos puu on liian syvä.

Ratkaisuna näihin ongelmiin on *satunnainen metsä* (*random forest*), jossa luodaan keskiarvotettu malli useammasta puusta. Opetusdatan syötedatasta \mathbf{x} ja ulostulodatasta \mathbf{y} otetaan satunnaisia osajoukkoja $\mathbf{x}_t \in \mathbf{x}, \mathbf{y}_t \in \mathbf{y}, t \in T$, missä t on puu ja T on puiden

joukko satunnaisessa metsässä. Merkitään puiden määrää $|T|$. Puiden määrän kasvaessa metsässä saadaan minimoitua ylisovituksen riski [1].

Nyt yksittäisten puiden mallit f_t luodaan normaaliin tapaan omilla opetusdatoilla. Tällöin ulostulo ennestään tuntemattomalle näytteelle \mathbf{x}' saadaan ottamalla keskiarvo kunkin puun f_t ulostulosta:

$$\hat{f}(\mathbf{x}') = \frac{1}{|T|} \sum_{t \in T} f_t(\mathbf{x}'), \quad (2.8)$$

missä $\hat{f}(\mathbf{x}')$ on koko metsän ulostulo. Kun kyseessä on luokittelupuu, valitaan luokka, joka on saanut eniten ulostuloja puista f_t . Tasapelitilanteessa ulostulo oletuksena arvotaan. [2, 3, 13]

Satunnaisen metsän ennusteen epävarmuudelle pystytään luomaan myös hyvä estimaatti. Otetaan kaikkien puiden ennusteiden keskihajonta:

$$\sigma = \sqrt{\frac{\sum_{t \in T} (f_t(\mathbf{x}') - \hat{f}(\mathbf{x}'))^2}{|T| - 1}}. \quad (2.9)$$

Tällöin siis mitä lähemmäksi kunkin puun ennuste osuu koko metsän keskiarvoa, sitä luotettavampi ennuste on. [2, 3]

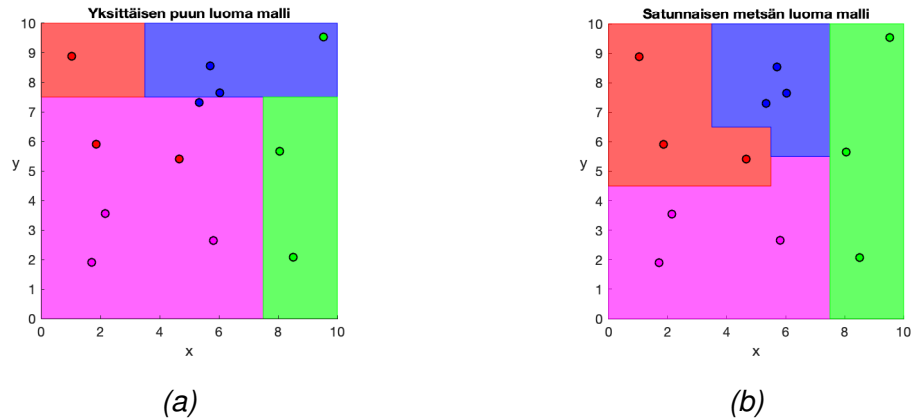
Satunnaisella metsällä on myös kätevä selvittää, mitkä parametrit isosta parametrijoukosta ovat hyödyllisimpiä. Määritellään muuttujan hyödyllisyys

$$\text{Imp}(x_j) = \frac{1}{|T|} \sum_{t \in T} \sum_{s \in t: v(s)=x_j} \frac{|\mathcal{D}_s|}{|\mathcal{D}|} \Delta(\mathcal{D}_s, s), \quad (2.10)$$

missä s on solmu, $v(s)$ on solmun jaossa käytetty muuttuja. Laskemalla hyödyn kaikille parametreille saadaan selville, mikäli jotkin parametrit ovat turhia muiden rinnalla. Nämä turhat parametrit voidaan jättää pois mallin luomisessa, sillä on riski, että yksittäiset puut ylisovittavat mallin näiden muuttujien osalta, mikä vaikuttaa aavistuksen koko puun suorituskykyyn. [3, 6, 12]

Satunnainen metsä on nykyisillä standardeilla tehokas ja nopea koneoppimisen metodi. Kun yksittäistä puuta painotetaan tarpeeksi vähän koko metsässä, minimoidaan ylisovituksen riski. Ja koska metsän avulla pystytään helposti selvittämään, millä parametreilla ei ole suurta vaikutusta ulostuloon, mallista pystytään poistamaan turhat parametrit. [2, 6]

Satunnaisen metsän pidemmälle viety tapaus on *extreme random forest*, missä puiden määrää kasvatetaan suuresti. Koska puun haarautumiseen tarvittavien optimaalisten ominaisuuksien ja kynnysarvojen laskeminen olisi liian raskasta, valitaan ne satunnaisesti. Tällöin puiden määrä voidaan nostaa jopa miljooniin, ja kukin puu pystyy käyttämään mallin luomiseen koko opetusdataa. [7]. Tässä diplomityössä ei kuitenkaan käytä kyseistä metodia.



Kuva 2.6. Esimerkki satunnaisen metsän toiminasta. Kuvissa ympyrät kuvaavat opetusdataa ja värilliset alueet mallin ulostuloa. Syötteenä on pisteen x - ja y -koordinaatti ja ulostulona pisteen väri. Kuvassa (a) on yksittäisen puun luoma malli, ja kuvassa (b) on 11 puusta muodostetun metsän malli.

Kuvassa 2.6 on esitetty yksinkertainen esimerkki satunnaisesta metsästä luokittelijana. Pallot kuvaavat mallien opetusdataa, jonka syötteenä on x - ja y -koordinaatti ja ulostulona pisteen väri. Värialueet kuvaavat mallien ulostuloa kullekin x - y -parille. Ensimmäisessä kuvassa on yksittäisen puun luoma malli ja toisessa kuvassa 11 puun keskiarvotettu malli. Vertailemalla kuvia (a) ja (b) voidaan havaita, että tämän yksittäisen puun malli ei vastaa niin hyvin opetusdataa kuin satunnaisen metsän malli. Ulostulon alueet eivät vastaa täysin opetusdataa varsinkaan violetin alueen osalta. Voidaankin olettaa, että tälle yksittäiselle puulle arvottu opetusdatan osajoukko koostuu suurelta osin violetista luokasta. Kuvasta (b) voidaan nähdä, että koko metsän malli vastaa kuitenkin opetusdatan tilannetta.

Tässä diplomityössä on käytetty *Python*-koodikielen *Scikit*-kirjaston toteutusta satunnaisesta metsästä. Luoduissa malleissa puiden määrä on ollut 70 puuta ja puiden suurin sallittu syvyys 2000 solmua. Muut parametrit ovat jätetty oletusarvoiksi.

3 TUTKIMUSONGELMA JA SEN TAUSTA

Satelliittipaikannuksella tarkoitetaan järjestelmää, jossa paikannussatelliittien avulla paikannuslaite pystyy passiivisesti laskemaan oman kolmiulotteisen sijaintinsa maapallolla. Paikan lisäksi laite saa tiedon omasta nopeudestaan sekä tarkan ajan kyseiseltä hetkeltä. Paikannussatelliitit kiertävät Maata noin 20 000 km etäisyydellä, ja tällä hetkellä kiertoradalla on yhteensä yli sata paikannussatelliittia useammasta eri satelliittikonstellatiosta. [5, 10, 18]

Tässä luvussa käydään läpi satelliittipaikannuksen matemaattinen periaate ja tutkimuksen motivaatio. Luvun lopussa on esitetty aiempia tutkimuksia aiheeseen liittyen.

3.1 Satelliittipaikannuksen periaate

Satelliittipaikannus perustuu satelliittien ja käyttäjän välisiin *pseudoetäisyysmittauksiin*. Satelliitit lähettävät signaalia, jonka saapumisaikaa vastaanotin tarkkailee. Signaali itsessään sisältää lähettämisaikakohdan. Koska käyttäjän kello on oletettavasti eri ajassa kuin satelliitin kello, ei voida tietää satelliitin ja käyttäjän välistä etäisyyttä vielä näillä tiedoilla. Pystytään kuitenkin laskemaan satelliitin ja käyttäjän välinen pseudoetäisyys:

$$\rho = r + c\Delta t + \varepsilon, \quad (3.1)$$

missä ρ on pseudoetäisyys, r on satelliitin ja käyttäjän välinen oikea etäisyys, c on valonnopeus, Δt on käyttäjän kellon virhe suhteessa satelliitin kelloon ja ε muut virhetermit. Voidaan kirjoittaa

$$b = c\Delta t, \quad (3.2)$$

jolloin b on kellon virheestä aiheutuva etäisyys pseudoetäisyyteen (*bias*). Yhdistämällä kaavat (3.1) ja (3.2) saadaan:

$$\rho = r + b + \varepsilon, \quad (3.3)$$

Saadaan siis 4 tuntematonta muuttujaa:

- käyttäjän x -koordinaatti,
- käyttäjän y -koordinaatti,
- käyttäjän z -koordinaatti,

- käyttäjän kellosta johtuva etäisyysmittauksen virhe b .

Yleisesti ei olla kiinnostuneita muuttujasta b , mutta se on silti ratkaistava, jotta saataisiin ratkaistua muut arvot. Koska tuntemattomia muuttujia on neljä, tarvitaan vähintään neljä pseudoetäisyysmittausta ongelman ratkaisemiseksi. Tällöin tarvitaan yhteydet neljään satelliittiin. [5, 9, 10, 18]

Merkitään käyttäjän paikkaa kolmiulotteisessa koordinaatistossa

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

ja satelliitin i paikkaa

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}.$$

Todellinen käyttäjän ja satelliitin i välinen etäisyys r_i on siis

$$r_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = \|\mathbf{x}_i - \mathbf{x}\|. \quad (3.4)$$

Yhdistämällä kaavat (3.3) ja (3.4) saadaan satelliitin i ja käyttäjän välinen todellinen pseudoetäisyys

$$\rho_i = \|\mathbf{x}_i - \mathbf{x}\| + b + \varepsilon_i. \quad (3.5)$$

Käytössä on siis vähintään neljältä satelliitilta pseudoetäisyydet, joista jokainen mallinnetaan epälineaarisella yhtälöllä (3.5). Pseudoetäisyydet linearisoidaan approksimoidulla käyttäjän sijainnilla \mathbf{x}_0 . Tämän avulla saadaan iteroitua parempi approksimaatio käyttäjän paikasta ja lopulta käyttäjän tarkka paikka. Alkuarvauksen ei tarvitse olla kovinkaan tarkka, mutta esimerkiksi jos alkuarvauksena on Maan keskipiste, ovat epälineaariset termit liian suuria, eikä iteraatio välttämättä suppene oikeaan paikkaan. [18] Tehdään vastaavasti termille b , mutta tämä voidaan arvioida aluksi nolllaksi. Saadaan

$$\rho_{i,0} = \|\mathbf{x}_i - \mathbf{x}_0\| + b_0, \quad (3.6)$$

missä $\rho_{i,0}$ on approksimoitu pseudoetäisyys, \mathbf{x}_0 on käyttäjän sijainnin alkuarvaus ja b_0 on kellosta johtuvan virheen alkuarvaus. Merkitään

$$\mathbf{x} = \mathbf{x}_0 + \delta\mathbf{x} \quad \text{ja} \quad b = b_0 + \delta b,$$

missä $\delta\mathbf{x}$ ja δb ovat mahdolliset parannukset nykyisiin approksimaatioihin. [5, 10, 18]

Nyt luodaan lineaaristen yhtälöiden ryhmän, josta pystytään ratkaisemaan muuttujat $\delta\mathbf{x}$

ja δb :

$$\begin{aligned}\delta\rho_i &= \rho_i - \rho_{i,0} \\ &= \|\mathbf{x}_i - \mathbf{x}_0 - \delta\mathbf{x}\| - \|\mathbf{x}_i - \mathbf{x}_0\| + (b - b_0) + \varepsilon_i \\ &\approx -\frac{\mathbf{x}_i - \mathbf{x}_0}{\|\mathbf{x}_i - \mathbf{x}_0\|} \cdot \delta\mathbf{x} + \delta b + \varepsilon_i,\end{aligned}\tag{3.7}$$

missä ollaan käytetty Taylorin sarjan approksimaatiota, ja $\frac{\mathbf{x}_i - \mathbf{x}_0}{\|\mathbf{x}_i - \mathbf{x}_0\|}$ on siis yksikkövektori käyttäjän arvioidusta sijainnista satelliitin i paikkaan. Yhtälöiden selkeyttämiseksi merkitään

$$\mathbf{1}_i = \frac{\mathbf{x}_i - \mathbf{x}_0}{\|\mathbf{x}_i - \mathbf{x}_0\|}.\tag{3.8}$$

Nyt yhtälöt voidaan kirjoittaa matriisimuodossa

$$\delta\boldsymbol{\rho} = \begin{bmatrix} \delta\rho_1 \\ \delta\rho_2 \\ \vdots \\ \delta\rho_n \end{bmatrix} = \begin{bmatrix} (-\mathbf{1}_1)^T & 1 \\ (-\mathbf{1}_2)^T & 1 \\ \vdots & \vdots \\ (-\mathbf{1}_n)^T & 1 \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta b \end{bmatrix} + \boldsymbol{\varepsilon},\tag{3.9}$$

missä n on niiden satelliittien lukumäärä, joista on tunnetut paikat ja mittaukset. Kun $n \geq 4$, voidaan ratkaista neljä tuntematonta muuttujaa: $\delta\mathbf{x}$ ja δb . Merkitään selkeyttämiseksi

$$\mathbf{G} = \begin{bmatrix} (-\mathbf{1}_1)^T & 1 \\ (-\mathbf{1}_2)^T & 1 \\ \vdots & \vdots \\ (-\mathbf{1}_n)^T & 1 \end{bmatrix},\tag{3.10}$$

missä \mathbf{G} on ns. *geometriamatriisi*. [5, 9, 10, 18]

Muuttujien yksikäsitteinen ratkaiseminen pienimmän neliösumman avulla vaatii, että matriisin \mathbf{G} sarakkeet ovat lineaarisesti riippumattomia. Mikäli matriisin \mathbf{G} sarakkeet ovat lineaarisesti riippumattomia, matriisi $\mathbf{G}^T\mathbf{G}$ on kääntövä. Vastaavasti jos matriisin sarakkeet ovat lineaarisesti riippuvia, matriisi $\mathbf{G}^T\mathbf{G}$ on singulaarinen. Neliömatriisi on ei-singulaarinen jos ja vain jos sen sarakevektorit ovat lineaarisesti riippumattomia [17]. Matriisi $\mathbf{G}^T\mathbf{G}$ on neliömatriisi, joten riittää todistaa, että matriisi $\mathbf{G}^T\mathbf{G}$ on kääntövä.

Todistus. Olkoon matriisin \mathbf{G} sarakkeet lineaarisesti riippumattomia. Tällöin yhtälölle

$$\mathbf{G}\underline{\alpha} = \underline{0}$$

on olemassa vain triviaaliratkaisu $\underline{\alpha} = \underline{0}$. Tutkitaan, ovatko neliömatriisin $\mathbf{G}^T\mathbf{G}$ sarakkeet

lineaarisesti riippumattomia:

$$\begin{aligned} \mathbf{G}^T \mathbf{G} \underline{\alpha} &= \underline{0} \\ \Rightarrow \underline{\alpha}^T \mathbf{G}^T \mathbf{G} \underline{\alpha} &= 0 \\ \Rightarrow \|\mathbf{G} \underline{\alpha}\|^2 &= 0 \\ \Rightarrow \mathbf{G} \underline{\alpha} &= \underline{0}. \end{aligned}$$

Oletuksen nojalla yhtälölle $\mathbf{G}^T \mathbf{G} \underline{\alpha} = \underline{0}$ on vain triviaaliratkaisu $\underline{\alpha} = \underline{0}$, joten matriisi $\mathbf{G}^T \mathbf{G}$ on kääntyvä. \square

Matriisi $\mathbf{G}^T \mathbf{G}$ ei ole kuitenkaan kääntyvä, jos kaikkien havaittujen satelliittien suuntavektorit pisteestä \mathbf{x}_0 ovat samassa tasossa, tai satelliitteja on alle neljä.

Todistus. Kun havaittuja satelliitteja on alle neljä, geometriamatriisi \mathbf{G} on muotoa

$$\mathbf{G} = \begin{bmatrix} (-\mathbf{1}_1)^T & 1 \\ \vdots & \\ (-\mathbf{1}_n)^T & 1 \end{bmatrix},$$

missä $n \leq 3$. Tällöin matriisin aste $\text{rank}(\mathbf{G}) = n \leq 3$, ja matriisilla on enintään $\text{rank}(\mathbf{G}) < 4$ lineaarisesti riippumatonta saraketta [17], jolloin matriisi $\mathbf{G}^T \mathbf{G}$ on singulaarinen.

Jos satelliittien suuntavektorit puolestaan ovat samassa tasossa, ne toteuttavat yhtälön

$$\underline{\beta} \mathbf{1}_1 = \underline{\beta} \mathbf{1}_2 = \dots = \underline{\beta} \mathbf{1}_n = d, \quad (3.11)$$

missä vakiovektori $\underline{\beta} \neq \underline{0}$ ja d on vakio [21]. Nyt geometriamatriisin \mathbf{G} sarakkeet ovat lineaarisesti riippuvaisia, sillä yhtälölle

$$\alpha_1 \begin{bmatrix} -\frac{x_1 - x_0}{\|\mathbf{x}_1 - \mathbf{x}_0\|} \\ -\frac{x_2 - x_0}{\|\mathbf{x}_2 - \mathbf{x}_0\|} \\ \vdots \\ -\frac{x_n - x_0}{\|\mathbf{x}_n - \mathbf{x}_0\|} \end{bmatrix} + \alpha_2 \begin{bmatrix} -\frac{y_1 - y_0}{\|\mathbf{x}_1 - \mathbf{x}_0\|} \\ -\frac{y_2 - y_0}{\|\mathbf{x}_2 - \mathbf{x}_0\|} \\ \vdots \\ -\frac{y_n - y_0}{\|\mathbf{x}_n - \mathbf{x}_0\|} \end{bmatrix} + \alpha_3 \begin{bmatrix} -\frac{z_1 - z_0}{\|\mathbf{x}_1 - \mathbf{x}_0\|} \\ -\frac{z_2 - z_0}{\|\mathbf{x}_2 - \mathbf{x}_0\|} \\ \vdots \\ -\frac{z_n - z_0}{\|\mathbf{x}_n - \mathbf{x}_0\|} \end{bmatrix} + \alpha_4 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \underline{0} \quad (3.12)$$

löytyy ei-triviaali ratkaisu $\alpha_i = \beta_i$, kun $i < 4$, ja $\alpha_4 = d$. Tällöin matriisi $\mathbf{G}^T \mathbf{G}$ on singulaarinen, eikä yksikäsitteistä pienimmän neliösumman ratkaisua löydy. \square

Satelliittien radat on asetettu siten, että ne peittävät maapallon mahdollisimman hyvin eri kulmista, joten yleisessä tapauksessa neljällä satelliitilla riippumattomuusoletus toteutuu, ja saadaan yksikäsitteinen pienimmän neliösumman ratkaisu [18]. Näin ei kuitenkaan aina ole. Kun satelliitit ovat pisteen \mathbf{x}_0 suhteen eri kulmissa, matriisin \mathbf{G} kolme ensimmäistä saraketta ovat lineaarisesti riippumattomat, mutta matriisin viimeinen sarake voi olla kolmen ensimmäisen sarakkeen lineaarikombinaatio.

Todistus. Vektorin \mathbf{x}_0 ollessa maanpinnalla koordinaatissa $61^\circ 26' 57.1''\text{N}$ $23^\circ 51' 21.3''\text{E}$ 5.1.2016 kello 00:11:32.885 UTC-aikaa, ja vastaanottimen saadessa yhteyden GPS-satelliitteihin PRN = 22, 23, 25 ja 31, niin geometriamatriisi \mathbf{G} on muotoa

$$\mathbf{G} = \begin{bmatrix} -0.4047 & -0.7205 & -0.5632 & 1 \\ -0.9382 & 0.0091 & -0.3461 & 1 \\ 0.7440 & -0.0472 & -0.6665 & 1 \\ 0.2104 & -0.7073 & -0.6749 & 1 \end{bmatrix}.$$

Suuntavektorit satelliitteihin eivät ole samassa tasossa, mutta matriisin determinantti $\det(\mathbf{G}) = 0$. Jos matriisin determinantti on 0, niin matriisin sarakkeet ovat lineaarisesti riippuvaisia [17], eli matriisi $\mathbf{G}^T \mathbf{G}$ on singulaarinen, eikä yksikäsitteistä pienimmän neliösumman ratkaisua löydy.

□

Lähes kaikissa tapauksissa matriisin \mathbf{G} sarakkeet ovat kuitenkin lineaarisesti riippumattomia, ja yksikäsitteinen pienimmän neliösumman ratkaisu löytyy. Ratkaisu voidaan kirjoittaa muodossa

$$\begin{bmatrix} \delta \mathbf{x} \\ \delta b \end{bmatrix} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \delta \boldsymbol{\rho}. \quad (3.13)$$

Nyt saadaan uudet tarkemmat estimaatit käyttäjän paikalle ja kellon aiheuttamalle virheelle:

$$\hat{\mathbf{x}} = \mathbf{x}_0 + \delta \mathbf{x} \quad \text{ja} \quad \hat{b} = b_0 + \delta b, \quad (3.14)$$

Uusilla paikka- ja virhe-estimaateilla voidaan iteroida aina entistä tarkemmat estimaatit. Iterointi luonnollisesti lopetetaan, kun kahden peräkkäisen iteraation estimaattien erotus on halutun pieni. [5, 10, 18]

Vastaavasti kolmella satelliitilla pystytään ratkaisemaan käyttäjän paikan sillä oletuksella, että käyttäjä on maan pinnalla. Jos menetetään yhteys joihinkin satelliittiin, pystytään silti jatkamaan paikannusta, kun oletetaan käyttäjän sijainnin pysyvän samana kadotetun satelliitin suhteen. Tällöin kuitenkin paikannustarkkuus kärsii oletettujen pseudoetäisyyksien takia. [10, 18] Tässä diplomityössä ei perehdytä syvällisemmin käyttäjän paikannukseen.

3.2 Tutkittava ongelma

Satelliittipaikannuksen toimivuuden lähtökohtana on, että tunnetaan satelliittien sijainnit avaruudessa. Satelliitit lähettävät tietoa omasta radastaan *Keplerin rataparametreilla*, joiden avulla käyttäjä pystyy laskemaan satelliitin sijainnin tarkasti seuraavista minuuteis-

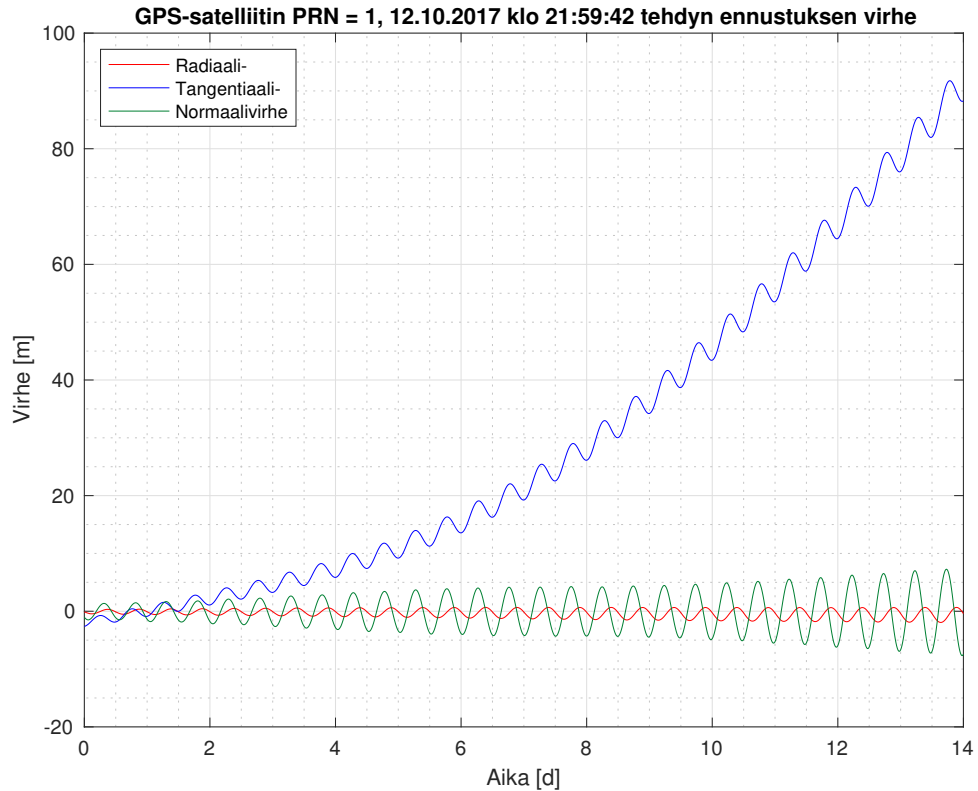
ta seuraaviin tunteihin konstellatiosta riippuen. Koska parametrit ovat käyttökelpoisia useamman hetken, lähettävät paikannussatelliitit samoja parametreja pidemmän aikaa. Esimerkiksi GPS-satelliitit lähettävät samoja parametreja kahden tunnin ajan ja GLONASS-satelliitit (*Globalnaya Navigatsionnaya Sputnikovaya Sistema*) 15 minuutin ajan. [10, 18]

Satelliitit lähettävät laitteille ratansa ohella myös kellokorjausparametreja, satelliitin terveystietoa, ionosfäärikorjausparametreja yms. samalla kaistalla. Tästä kaistasta GPS-satelliiteilla rataparametrien osuus on 12 sekuntia 30 sekunnista. Kun käyttäjä aloittaa paikantamisen, hänen on siis ensin odotettava, että hän saa vastaanotettua tarvittavien satelliittien ratatiedot. [8, 18] On myös mahdollista, että jonkin häiriön vuoksi käyttäjä ei onnistu vastaanottamaan jotain osaa parametreista, jolloin tämä joutuu odottamaan uudestaan 30 sekuntia, jotta saisi kaikki tarvittavat parametrit. Näitä häiriöitä aiheuttaa esimerkiksi rakennukset, suuret puut ja tunnelit, kun ne ovat satelliitin ja käyttäjän välissä vaimentaen signaalia. [9, 18]

Tätä odotusaikaa voidaan pienentää, jos on aiemmin ennustettu satelliittien paikat menneisyydessä ja tallennettu niiden tulevat radat laitteeseen. Tällöin paikannus voidaan aloittaa ennustettujen sijaintien avulla välittömästi. Alkupaikannuksen tarkkuus riippuu siitä, kuinka tarkasti ennustus satelliitin nykyisestä paikasta on onnistuttu tekemään.

Tässä diplomityössä on käytetty lähtökohtana ennustusmallia, joka ottaa huomioon Maan, Kuun ja Auringon vetovoiman sekä Auringon säteilypaineen [20]. Nimitetään tätä mallia *normaaliksi ennustusmalliksi* tästä eteenpäin. Mallilla on tarkoitus ennustaa jopa 14 päivän päähän satelliitin rataa. Ennustusajan kasvaessa ennustusvirhe alkaa luonnollisesti kasvamaan.

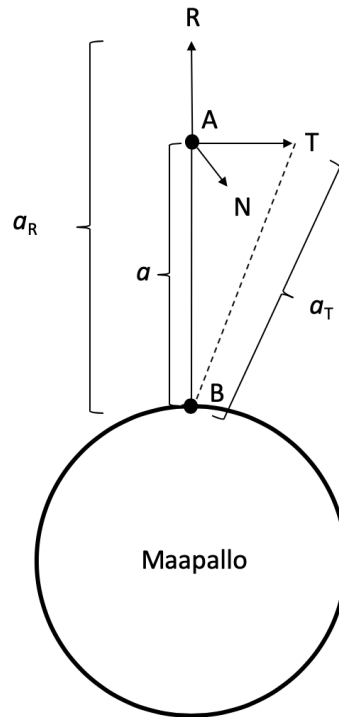
Kuvassa 3.1 on havainnollistettu yksittäisen virheen luonnetta. Koska satelliitilla on suuri nopeus radan tangentin suuntaan, noin 14 000 km/h, niin suurin osa virheestä kasautuu tangentiaalikomponenttiin. Radiaali- ja normaalikomponentit värähtelevät pääsääntöisesti kasvavalla amplitudilla nollan ympäristössä. Myös tangentiaalikomponentilla havaitaan samanlaista värähtelyä.



Kuva 3.1. Normaalilla ennustusmallilla tehdyn ennustuksen virhe komponenteittain.

Kaikki komponentit eivät kuitenkaan aiheuta yhtä suurta virhettä paikannustarkkuuteen. Radiaalivirhe on suurin virheen aiheuttaja, sillä radiaalikomponentti on lähes samansuuntainen käyttäjän ja satelliitin välisen etäisyysvektorin kanssa. Tällöin radiaalivirhe aiheuttaa lähes itsensä suuruisen virheen etäisyysmittaukseen. Tangentiaali- ja normaalikomponentit puolestaan ovat lähes kohtisuoraan etäisyysvektorin suhteen.

Kuvassa 3.2 on havainnollistettu ilmiötä. Olkoon A satelliitin ja B vastaanottimen paikka, a satelliitin ja käyttäjän välinen todellinen etäisyys, a_R satelliitin radiaalivirheestä johtuvan virheellisen paikan ja käyttäjän välinen etäisyys ja a_T vastaava etäisyys tangentiaalivirheelle. Todellisuudessa satelliitti ei ole jatkuvasti käyttäjän zeniitissä. Satelliitin ja käyttäjän välinen todellinen etäisyys vaihtelee noin 20 000 kilometristä noin 25 000 kilometriin. Virhekomponenttien ollessa kahdesta neljään kertaluokkaa pienempiä ei tangentiaali- tai normaalivirhe vaikuta suuresti etäisyysmittaukseen.



Kuva 3.2. Kuva havainnollistaa radiaali-, normaali- ja tangentialivirheen vaikutusta etäisyyssmittaukseen. Piste A kuvastaa satelliittia ja piste B vastaanotinta. Kuva ei ole oikeassa mittakaavassa.

Näiden virhekomponenttien aiheuttamaa keskiarvoista paikannusvirhettä Maan päällä voidaan kuvata SISRE:llä (*Signal-In-Space Range Error*). SISRE määritellään kullekin konstellaatiolle erikseen, sillä satelliitin radan säde vaikuttaa virhekomponenttien osuuksiin paikannusvirheessä. SISRE:n kaava on:

$$\text{SISRE} = \sqrt{w_R^2 \Delta R^2 + w_{T,N}^2 (\Delta T^2 + \Delta N^2)}, \quad (3.15)$$

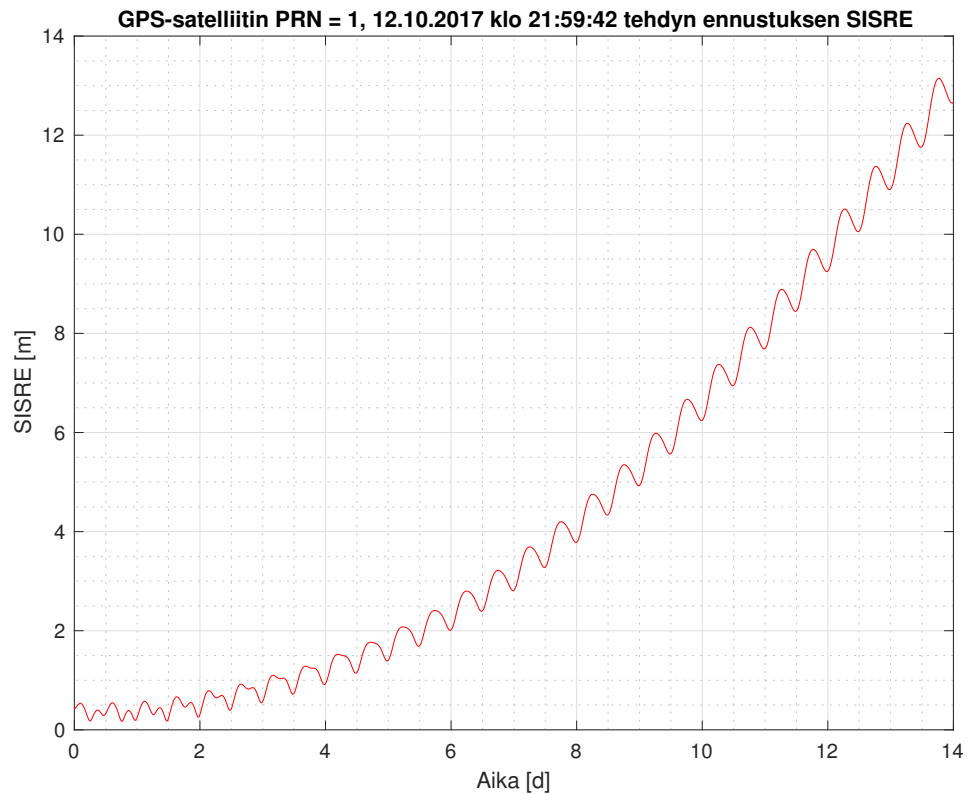
missä ΔR on radiaalivirhe, ΔT tangentialivirhe, ΔN normaalivirhe, w_R radiaanivirheen kerroin ja $w_{T,N}$ tangentiali- ja normaalivirheen kerroin. w_R ja $w_{T,N}$ riippuvat satelliittien radan säteestä. Taulukossa 3.1 on määritetty muutamien satelliittityyppien vakioiden arvot. [14]

	GPS	GLONASS	BeiDou MEO	BeiDou IGSO/GEO
w_R	0,98	0,98	0,98	0,99
$w_{T,N}^2$	$\frac{1}{49}$	$\frac{1}{45}$	$\frac{1}{54}$	$\frac{1}{126}$

Taulukko 3.1. Vakioiden w_R ja $w_{T,N}^2$ arvot muutamilla satelliittityypeillä. Arvot on kerätty lähteestä [14].

Tässä diplomityössä testataan korjausmallin toimivuutta vain GPS-satelliiteilla. Mainitakoon, että taulukossa 3.1 mainittu GLONASS on venäläinen globaali satelliittipaikan-

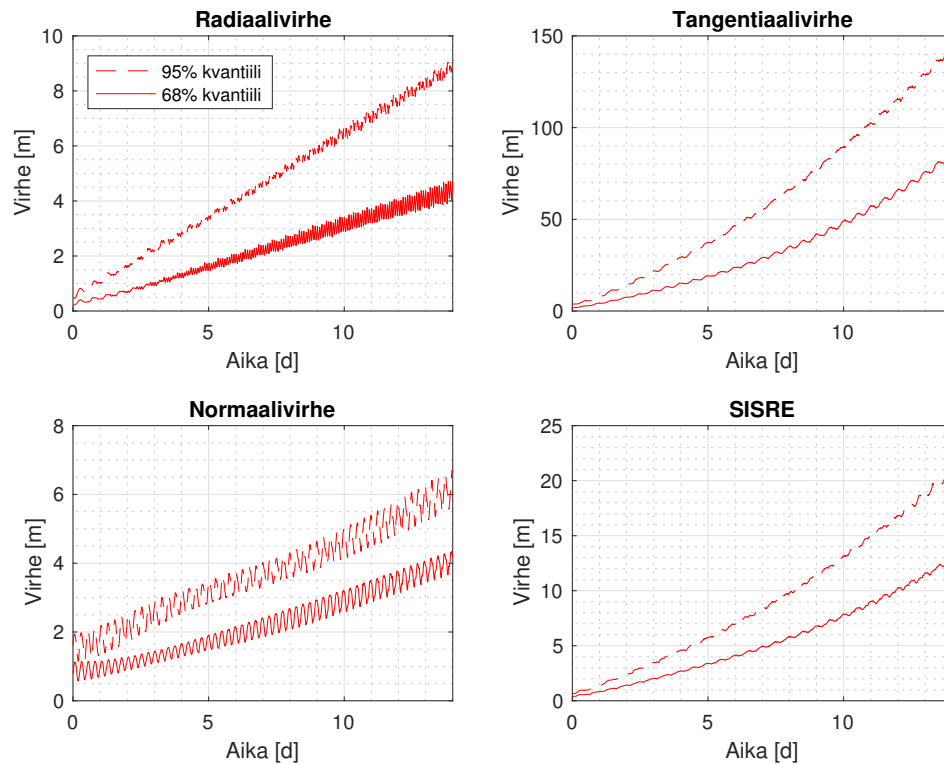
nusjärjestelmä, ja *BeiDou* on kiinalainen globaali satelliittipaikannusjärjestelmä, jota on täydennetty lokaaleilla IGSO-(*Inclined Geosynchronous Orbit*) ja GEO-satelliiteilla (*Geosynchronous satellite*). MEO-satelliitit (*Medium Earth Orbit*) puolestaan ovat globaaleja.



Kuva 3.3. Normaalilla ennustusmallilla tehdyn ennustuksen SISRE.

Kuvassa 3.3 on esitetty kuvan 3.1 virhekomponenttien muodostama SISRE. Vaikka tangentiaalivirheen osuus SISRE:stä on huomattavasti radiaalivirheen osuutta pienempi, se on vaikuttavin tekijä suuruutensa tähden. Muutaman päivän jälkeen SISRE noudattaakin pitkälti tangentiaalivirheen muotoista käyrää. On kuitenkin hyvä huomata, että tangentiaalivirheen ollessa 14. päivän kohdalla noin 90 metriä kuvassa 3.4, niin sitä vastaava SISRE on vain noin 13 metriä, niin kuin kuvasta 3.3 voidaan havaita.

RTN-virhe ja SISRE, PRN 1, 7.10.2017 - 22.5.2018



Kuva 3.4. Normaalin ennustusmallin virhekomponenttien itseisarvon ja SISRE:n 95 %- ja 68 %-kvantiilit noin seitsemän kuukauden ajalta.

Edelliset kuvat edustavat kuitenkin vain yksittäistapausta. Kuvassa 3.4 esitetään noin seitsemän kuukauden ajanjaksolta laskettujen 14 päivän mittaisten ennustusten virhekäyrien itseisarvojen sekä niitä vastaavien SISRE:jen 95 %- ja 68 %-kvantiileja. Tästä voidaan havaita, että SISRE pysyy vielä 14 päivän ennustuksessa yleisesti alle 20 metrin, mikä on siedettävä virhe kahden viikon ennustukselta.

Tässä diplomityössä tarkoitus on parantaa normaalia ennustusmallia koneoppimisen avulla. Lähtökohtana on, että paikannuslaite pystyisi tekemään ennustamisen itsenäisesti ilman jatkuvaa palvelimen tukea. Laite pystyy kuitenkin kiirettömästi keräämään palvelimella laskettuja avusteparametreja satunnaisina aikoina, esimerkiksi kun laite on WLAN-yhteydessä (*Wireless Local Area Network*).

Ratkaisun kannalta on siis tärkeää, että parametreja pystytään laskemaan ja lähettämään laitteelle tarpeeksi ajoissa, tai että parametrien elinikä on tarpeeksi pitkä. Myös lähetettävien parametrien tiedostokoko tulisi huomioida.

3.3 Aiemmat tutkimukset

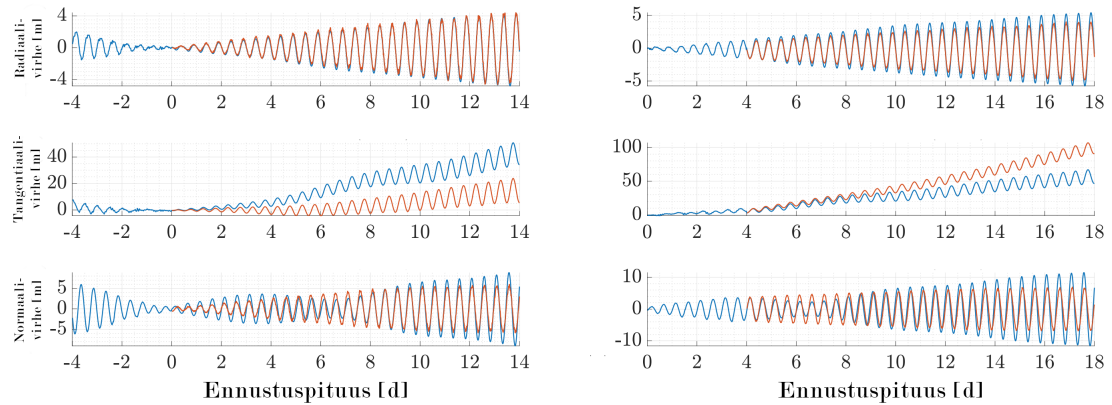
Satelliittien rataennustusten suhteen on tehty vuosien saatossa tutkimusta eri näkökulmista. Mallin parantaminen on kuitenkin ollut vaivalloista pelkän fysiikan avulla. Varsinkin auringon säteilypaineen tarkka mallinnus on vaikeaa. Aihetta on lähestytty esimerkiksi heuristisella menetelmällä, jossa tarkemman mallin sijaan pyritään mallintamaan normaalin ennustusmallin virhettä neuroverkon avulla. Tällöin kun rataennustuksesta vähennetään ennustettu virhe, saadaan parannettua ennustusta. [16]

Pihlajasalon diplomityössä [16] käytettiin hyväksi mahdollisuutta, jossa pystytään internetin välityksellä lataamaan ennustushetkeä edeltäviä BE:tä useamman päivän takaa. Näin pystytään tekemään simuloitu ennustus menneisyydestä, vertaamaan sen onnistumista, ja sen perusteella tekemään korjaus nykyiseen ennustukseen. Parempiin tuloksiin päästiin kuitenkin tekemällä mallinnus satelliitin radasta taaksepäin, vertaamalla mallinnettua rataa todelliseen historiatietoon, ja tekemällä korjaus oikeaan ennustukseen tämän perusteella. Kunkin ennustuksen korjausparametrit laskettaisiin palvelimella ja lähetettäisiin paikannuslaitteisiin internetin välityksellä.

Korjaus on tehty yksinkertaisen konvoluutioverkon avulla, jossa verkon syöte on tunnetun ennustuksen RTN-virhe, ja ulostulona uuden ennustuksen RTN-virhe. RTN-virhe oli ilmoitettu RGB-kuvina, jossa kukin värikanava kuvaa yhtä virhekomponenttia. Opetukseen käytettiin muutamien kuukausien simuloituja ennustuksia jokaiselta satelliitilta.

Tällä menetelmällä saatiin selkeitä parannuksia sekä GPS- että BeiDoun IGSO- ja GEO-satelliittien rataennustuksiin. Parannus oli kahden viikon ennustuksilla keskimäärin 40 % kaikilta satelliiteilta. Parhaat tulokset saatiin kuitenkin BeiDou-satelliiteilla. [16] Kuten kuvasta 3.5 nähdään, konvoluutiolla tehdy korjausmalli voi ennustaa normaalin ennustusmallin ennustusvirhettä todella hyvin.

Menetelmän heikkoutena kuitenkin on, että joskus taaksepäin tehdyllä ennustuksella luotu korjaus poikkeaa todellisesta virheestä selkeästi. Tähän on kuitenkin kyetty luomaan tunnistusalgoritmi, joka kykenee tunnistamaan nämä mahdolliset tapaukset hyvällä osu-matarkkuudella. Kun mahdollinen huono korjaus tunnistetaan, käytetään normaalia ennustusmallia ilman korjauksia.



Kuva 3.5. Konvoluutioverkolla ennustettu virhe GPS-satelliitille PRN=23 virhekomponentteittain. Kuvaajat ovat ylhäältä alaspäin radiaali-, tangentiaali- ja normaalivirheet. Vasemmanpuolisissa kuvaajissa korjaus on tehty taaksepäin ja oikeanpuolisissa eteenpäin tehtyjen ennustusten avulla. Molemmissa on käytetty 4 päivän ennustusdataa syötteenä. Sininen viiva kuvastaa todellista virhettä ja punainen mallinnettua korjausta. Kuva muokattu lähteestä [16].

Peng ja Bai ovat tutkineet koneoppimisen käyttöä avaruuden pienten kappaleiden kiertoratojen ennustukseen [15]. Periaate on samankaltainen: puutteellista fysikaalista mallia pyrittiin korjaamaan tukivektorikoneen avulla tarkemmaksi.

Mallin opetuksessa käytettiin hyväksi saman kappaleen aikaisempaa rataa sekä lähekkäisten kappaleiden ratoja. Mallin syöteinä olivat edellisten ennustusten suhteellinen virhe, ennustuksen pituus, ennustetun radan muoto, estimoitu ilmanvastus, kappaleen korkeus- ja suuntakulma sekä etäisyys Maahan. Mallin ulostulona oli ennustuksen virhe. Korjausmallilla kyettiin noin puolittamaan kappaleiden rataennustuksen virhe. [15]

4 MALLIN LUOMINEN

4.1 Parametrien valinta

Koneoppimista käytettäessä ongelma on se, mitä parametreja annetaan koneelle syöteenä, jotta saataisiin parhas mahdollinen lopputulos. Syöteen ja ulostulon välillä tulisi olla jonkinlainen relaatio, josta kone tekee korjausmallin. Yksinkertaisimmissa tapauksissa käytössä on vain muutama mahdollinen syöte ja optimaalisten syötteiden valinta onnistuu kokeilemalla kaikki mahdolliset syötteiden osajoukot.

Koneelle annetut "turhat" parametrit eivät yleensä vaikuta suuresti mallin tarkkuuteen. Satunnainen metsä kuitenkin pyrkii löytämään kaikkiin parametreihin sopivan mallin. Tällöin on mahdollisuus siihen, että malli on ylisovitettu parametreihin, jolla ei ole mitään todellista relaatiota tuloksen kanssa. Tällä voi huonontaa mallin tarkkuutta.

Kuten kappaleessa 2.2.3 todettiin, satunnaisen metsän avulla pystytään selvittämään, mitkä parametrit ovat muita hyödyttömämpiä mallin luomisessa. Tässä tutkimuksessa kokeillaan kuitenkin varmuuden vuoksi tehdä useampi malli, joista kustakin on poistettu jokin parametrisetti. Tällöin saadaan selville se, kuinka paljon kukin parametrisetti parantaa mallia.

Tässä diplomityössä keskitytään ongelmaan, jossa pyritään ennustamaan satelliitin radan virhettä yhden lähetysefemeridin (BE) avulla. Tätä varten on kerätty parametreja, joilla on itsessään rooli satelliitin radan ennustuksessa. Tämän lisäksi parametreiksi on otettu aika eri muodoissa. Parametrit on esitetty taulukossa 4.1. Tässä diplomityössä pyritään kuitenkin mallintamaan normaalin ennustusmallin virhettä, minkä takia kaikilla taulukon parametreilla ei tule oletettavasti olemaan vaikutusta ennustuksen virheeseen.

Satelliitin PRN viittaa satelliitin tunnistenumeroon. Jokaisella satelliitilla on oma virheensä, ja siksi koneelle annetaan syöteenä satelliitin tunnistenumero kategorisesti. Testien nojalla tämä parametri vaikuttaa suuresti korjausmallin toimivuuteen. Yksi vaihtoehtoinen lisäparametri olisi satelliitin rata. GPS-satelliitit kulkevat vain muutamalla eri radalla. Kaikkien ratojen säde on sama, mutta kallistuskulma eri. Samalla radalla kulkevat satelliitit kulkevat samalla nopeudella ja eri vaiheessa.

Paikan säde sekä suunta- ja korkeuskulmat vaikuttavat suuresti normaalissa ennustusmallissa. Ennustukset pitää sovittaa nykyiseen paikkaan ja paikan säde vaikuttaa satelliitin vaikuttaviin gravitaatiovoimiin. Kuitenkaan korjausmallin toimivuus ei hyödy satelliitin

Attribuutti	Arvo
Satelliitin PRN	kategorinen
Paikan säde	liukuluku
Paikan suuntakulma	liukuluku $[-\frac{\pi}{2}, \frac{\pi}{2}]$
Paikan korkeuskulma	liukuluku $[-\pi, \pi]$
Nopeuden säde	liukuluku
Nopeuden suuntakulma	liukuluku $[-\frac{\pi}{2}, \frac{\pi}{2}]$
Nopeuden korkeuskulma	liukuluku $[-\pi, \pi]$
Auringon valossa	Binaarinen, 0 = varjossa, 1 valossa.
Vuodenaika	liukuluku $[0,1]$
Vuorokaudenaika	liukuluku $[0,1]$
GPS-sekunti	luonnollinen luku

Taulukko 4.1. Tutkittavat parametrit

paikan tiedoista, eli paikalla ei ole vaikutusta ennustuksen tarkkuuteen.

Sama ilmiö huomataan nopeuden säteessä sekä suunta- ja korkeuskulmissa. Yksinkertainen voimamalli käyttäisi radan ennustamiseen vain sen paikkaa ja nopeutta sekä Maan, Kuun ja Auringon massaa ja paikkaa. Kuitenkaan korjausmalli ei hyödy satelliitin nopeudesta syötteenä. Voidaankin siis päätellä, että mallin virheet eivät johdu satelliitin paikasta tai nopeudesta.

Auringon säteilypainne on yksi suurimmista voimista, joka vaikuttaa satelliittiin. Auringosta lähtevät fotonit työntävät satelliittia pois päin Auringosta. Kuvataan tätä binäärisellä arvolla, joka saa arvon 1, kun satelliitti on Auringon valossa, ja arvon 0, kun satelliitti on Maan varjossa. Ongelmaksi tulee kuitenkin ennustuksen pituus. Tarkoituksena on ennustaa satelliitin rataa kahden viikon ajalta. GPS-satelliitti kulkee maapallon ympäri noin kaksi kertaa vuorokaudessa, jolloin Auringon valon osuminen satelliittiin ehtii muuttua kymmeniä kertoja ennustuksen aikana. Vaikuttaa siltä, ettei yksittäinen binääriparametri pysty kuvaamaan säteilypainetta tarpeeksi hyvin. Tällä parametrilla ei siis ole vaikutusta korjausmallin ennustustarkkuuteen. Säteilypainne on kuitenkin arvioitu olemaan suurin virheen

lähteistä [11], joten tulevaisuudessa tutkimuksissa tulisi pohtia, saisiko säteilypaineen annettuna jotenkin hyödyllisempänä parametrina korjausmallille.

Parametrien kokeilussa havaittiin, että tangentiaali-, radiaali- ja normaalivirheellä on selkeä aikariippuvuus. Tangentiaalivirhe on siniaaltoja muistuttava vuoden jaksonajalla kulkeva käyrä. Ennustuspituuden kasvaessa kasvaa myös virheen amplitudi.

Normaali- ja radiaalivirheen jaksonaika on selkeästi lyhyempi, noin puoli vuorokautta. Tämä johtuu siitä, että GPS-satelliittien kiertoaika on 11 tuntia ja 58 minuuttia. Virheet kasautuvat, kun satelliitti kulkee radan samoissa kohdissa uudestaan. Virheen amplitudi kasvaa tangentiaalivirheen tavoin ennustuspituuden mukaan.

Tullaan siis tulokseen, että korjausmalli opetetaan seuraavilla parametreilla: satelliitin PRN-numero, vuodenaika, vuorokaudenaika ja GPS-sekunti. Nämä neljä parametria ovat myös määritelmän (2.10) mukaan hyödyllisimmät parametrit.

Attribuutti	Arvo
Satelliitin PRN	kategorinen
Vuodenaika	liukuluku [0,1]
Vuorokaudenaika	liukuluku [0,1]
GPS-sekunti	luonnollinen luku

Taulukko 4.2. Syötteet, joilla malli opetetaan.

4.2 Mallin ulostulo

Kuten kappaleessa 3 todetaan, tarkoituksena ei ole satunnaisen metsän avulla suoraan ennustaa satelliitin rataa, vaan luoda korjausparametreja normaalille ennustusmallille. Selvitettäväksi jää, minkälaiset parametrit halutaan korjausmallista ulos, sillä koneoppimisessa syöteparametrien ohella tarkkuutta voidaan parantaa mallin ulostulon valitsemisella. Vaikka ulostulo kuvaisi oikeaa dataa hyvin, se voi olla toisaalta vaikeasti mallinnettavissa koneoppimisen avulla. Tällöin mallin toimivuutta voidaan parantaa yksinkertaistamalla ulostuloa, jolloin malli ei täysin vastaa oikeaa dataa, mutta on helpommin mallinnettavissa.

Jokaiselle kolmelle virhekomponentille luodaan oman "metsän". Radiaali- ja normaalivirheen samanlaisesta luonteesta johtuen niiden metsät luodaan samaa periaatetta käyttäen. Metsien koko ja syvyys ovat samat, kuten myös mallin ulostulon parametrit.

Kaikki virhekomponenttien parametrit voitaisiin ennustaa myös samalla metsällä, mutta

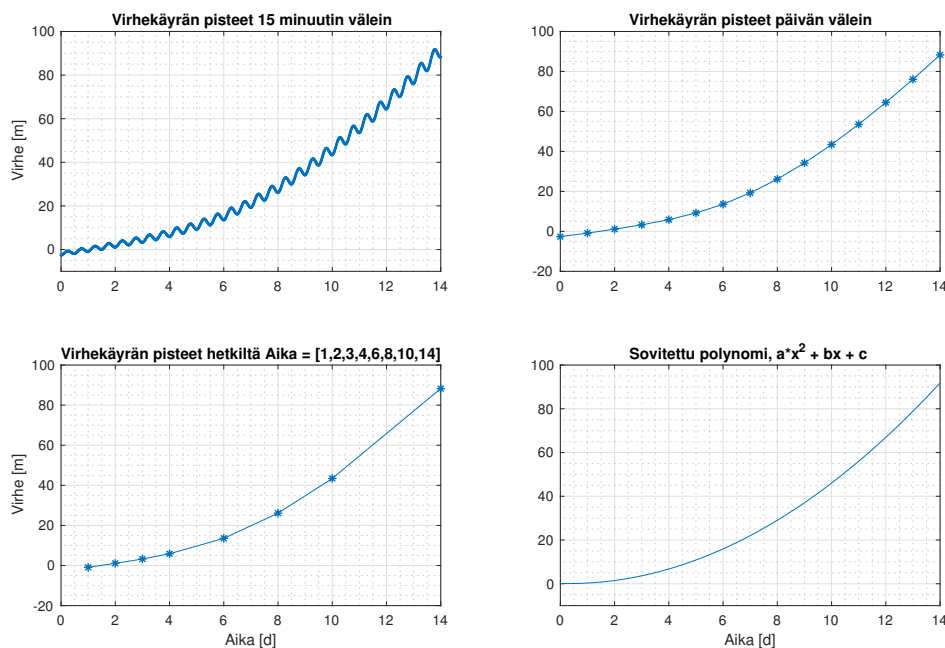
ennustustarkkuus paranee, kun jokaisella on oma metsänsä. Vaikuttaa siltä, että samassa mallissa metsä pyrkii parantamaan tangentiaalivirheen ennustamista radiaali- ja normaalivirheen kustannuksella. Jos kuitenkin tarve olisi tuottaa kaikki parametrit samalla metsällä, tulisi ulostulon parametrit skaalata samalle tasolle, jotta välttyttäisiin tangentiaalivirheen ennustamisen painottumiselta. Tätä ei kuitenkaan olla nähty tarpeelliseksi, joten on luotu kolme eri metsää.

4.2.1 Tangentiaalivirhe

Tangentiaalivirhettä voidaan yleisesti kuvata toisen asteen polynomina, johon on lisätty siniaalto kasvavalla amplitudilla. Parhaiten virhettä kuvaasi ulostulo, jossa otetaan virhe-
käyrältä pisteitä tiheästi tasaisin väliajoin, esimerkiksi 15 minuutin välein. Pisteiden välit saataisiin yksinkertaisesti lineaarisella tai epälineaarisella sovituksella. Tällöin kuitenkin mallin ulostulo koostuisi $1 + 14 \cdot 24 \cdot 4 = 1345$ pisteestä (14 päivää, 24 tuntia päivässä, 4 pistettä tunnissa ja lisäksi aloituspiste). Mallista tulisi näin ollen monimutkainen, ja sen ennustustarkkuudessa olisi parannettavaa.

Kuvassa 4.1 on esitetty muutama tapa pienentää parametrien määrää. Ensimmäisessä kuvassa on esitetty geneerinen tangentiaalivirhe 15 minuutin näytteenottotaajuudella, toisessa yhden päivän näytteenottotaajuudella ja kolmannessa näytteenottopisteitä on alussa tiheämmin kuin lopussa. Neljännessä kuvassa puolestaan ensimmäisen kuvan dataan on sovitettu toisen asteen polynomi.

Huomataan, että näytteenottovälin harventuessa menetetään tieto tangentiaalivirheelle tyypillisestä sinivärähtelystä. Tämä kuitenkin on valinta, jonka voidaan tehdä, sillä kun mallia saadaan yksinkertaistettua, sen ennustustarkkuus paranee. Näin malli kokonaisuudessaan ennustaa tarkemmin tangentiaalivirhettä. Tangentiaalivirheen pienistä parannuksista ei sitä paitsi tuo paljoa parannusta paikannustarkkuuteen, mikä voidaan havaita kaavasta (3.15).



Kuva 4.1. Tangentiaalivirheen kuvaus eri parametrimäärillä.

Sovittamalla virhekäyrään toisen asteen polynomin saadaan parhaat tulokset. Lisäksi toisen asteen käyrän muodostamiseen tarvitaan vain kolme parametria. Tämä vähentää suuresti laitteelle lähetettävien parametrien määrää alkuperäiseen 1345 parametriin verrattuna. Vielä yksi lähetettävä parametrin saadaan vähennettyä, kun oletetaan, että käyrän $ax^2 + bx + c$ parametri $c = 0$, sillä virhekäyrän voidaan olettaa lähtevän origosta.

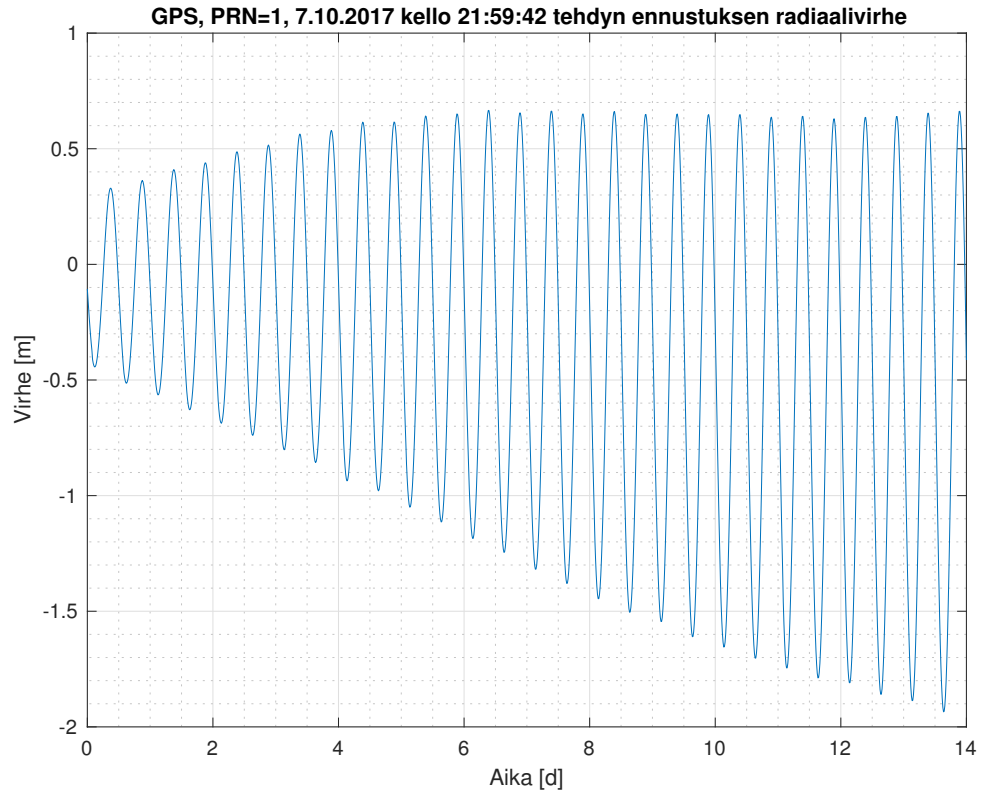
Tangentiaalivirheen ulostuloksi saadaan siis yksinkertaisuudessaan taulukon 4.3 mukaiset parametrit:

Attribuutti	Arvo
Toisen asteen polynomin parametri a	liukuluku
Toisen asteen polynomin parametri b	liukuluku

Taulukko 4.3. Tangentiaalivirhettä ennustavan mallin ulostulo.

4.2.2 Radiaali- ja normaalivirhe

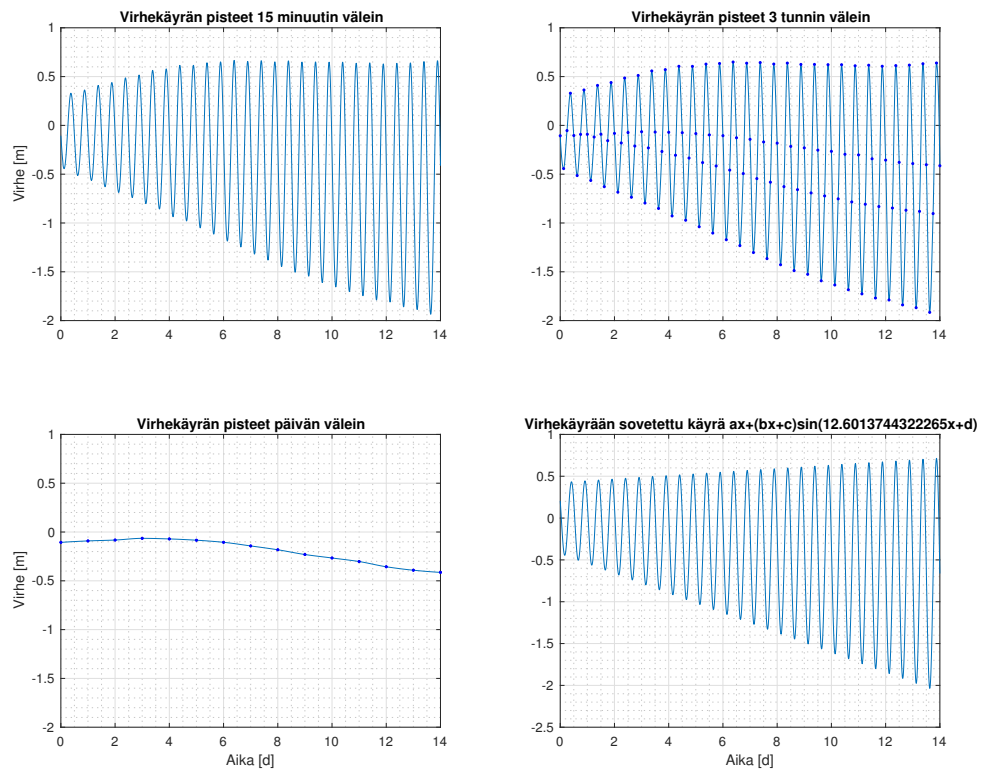
Radiaali- ja normaalivirheet ovat luonteeltaan samanlaisia. Yleisessä tapauksessa ne värähtelevät kasvavalla amplitudilla nollan ympärillä. Kasvava värähtely muodostuu, kun radan virheet kumuloituvat radan samassa kohdassa. Koska GPS-satelliitilla kuluu noin 12 tuntia kiertää maapallo ympäri, on värähtelyn jaksonaika noin 12 tuntia. Värähtelyn nol-lakeskeisyys ei kuitenkaan ole taattua, niin kuin kuvasta 4.3 voidaan havaita. Virhekäyrä voi siis poiketa nollasta joko negatiiviseen tai positiiviseen suuntaan.



Kuva 4.2. Esimerkki radiaalivirheen käyttäytymisestä.

Radiaali- ja normaalivirheen tapauksessa tulee myös tutkia, mikä on optimaalinen määrä ulostulon arvoja. Kuvassa 4.3 on havainnollistettu neljää eri tapausta ulostulon arvoista. Ensimmäisessä kuvassa ulostulona on virhekäyrä 15 minuutin näytteenottotaajuudella. Toisessa kuvassa näytteenottotaajuus on pudotettu 3 tuntiin. Vahvemmat siniset pisteet kuvaavat näytteitä. Voidaankin havaita, että näytteenottotaajuuden pudottamisella tähän asti ei näyttäisi olevan vaikutusta tarkkuuteen. Tällöin parametrien määrä on $1 + 8 \cdot 14 = 113$.

Kolmannessa kuvassa näytteenottotaajuutta on pudotettu radikaalisti 24 tuntiin. Vaikka tangentiaalivirheellä tämä ei vielä aiheuttanut paljoa informaation menetystä, ei radiaalivirheen tapauksessa enää alkuperäistä käyrää pystytä näillä mittauspisteillä luomaan. Radiaali- ja normaalivirheellä ei siis tällä lähestymistavalla pystytä vähentämään ulostulon arvoja.



Kuva 4.3. Radiaalivirheen kuvaus eri parametrimäärillä.

Neljännessä kuvassa virhekäyrään on sovitettu käyrä

$$ax + (bx + c) \cdot \sin(12.6013744322265 + d), \quad (4.1)$$

missä x on aika päivinä. Vakio 12.6013744322265 saadaan, kun tiedetään GPS-satelliitin kiertoajan olevan 11 tuntia ja 58 minuuttia, josta saadaan:

$$\frac{12 \text{ tuntia}}{11 \text{ tuntia ja } 58 \text{ minuuttia}} \cdot 4\pi \approx 12.6013744322265.$$

Tällä keinolla ei kuitenkaan saada parempia tuloksia, sillä käyrän parametrien muuttamista ajan funktiona on vaikea mallintaa luotettavasti koneoppimisen avulla. Parhaaksi keinoksi osoittautui ennustaa käyrän pisteitä 3 tunnin välein ja sovittaa niihin interpoloiva polynomi PCHIP (*Piecewise Cubic Hermite Interpolating Polynomial*). Käyrän parametrit voidaan kuitenkin lähettää laitteelle käyrän (4.1) muodossa, jolloin tarkkuus huononee, mutta lähetettävien parametrien määrä pienenee 113:sta neljään.

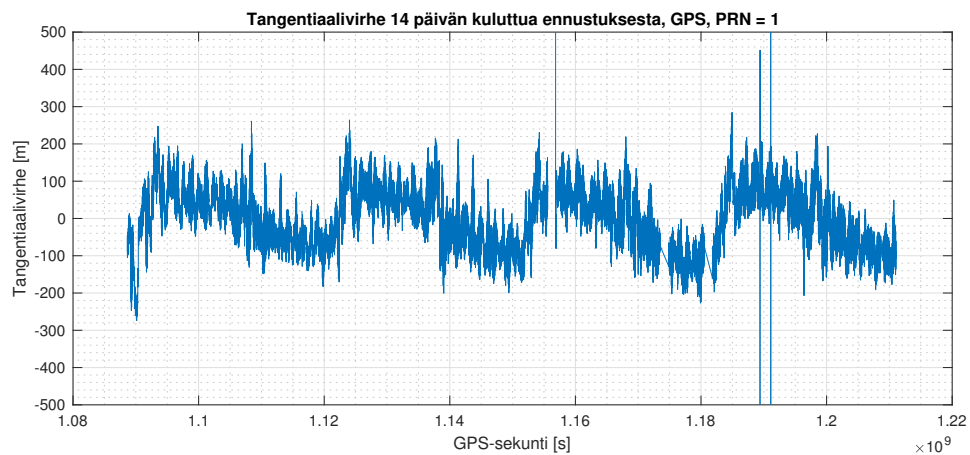
4.3 Datan esikäsittely

Satunnainen metsä pyrkii mallintamaan syötedatan ulostuloa sen perusteella, mitä ulostuloa sille on opetettu. Todelliselle datalle on kuitenkin tyypillistä, että se sisältää kohinaa,

sekä syötedatasta riippumattomia ja muusta datasta erottuvia poikkeamia. Luonnollises-
tikan näiden käyttäytymistä ei pystytä mallintamaan.

Satunnaisen metsän yksittäiset puut kuitenkin pyrkivät löytämään kaiken säännöllisyy-
den opetusdatasta. Tällöin datassa voi esiintyä kohinaa ja poikkeamia, joille pystytään
opetusdatasta löytämään monimutkainen säännöllisyys, joka ei kuitenkaan päde yleises-
sä tapauksessa. Vaikka yksittäinen puu ei vaikutakaan suuresti koko metsän suoritus-
kykyyn, mallia voidaan parantaa hieman poistamalla opetusdatasta liiallinen kohina ja
poikkeamat.

Kuvassa 4.4 on havainnollistellut, miltä tangentiaalivirhe 14 päivän pituisilla ennus-
tuksilla näyttää useamman vuoden jaksolla. Kuten kuvasta näkyy, datassa on satunnai-
sesti isompia piikkejä. Nämä ovat tapahtumia, jolloin satelliitin rataa korjataan polttoai-
neavusteisesti, jolloin sen rata muuttuu huomattavasti. Tällöin ennustus menee selkeästi
pieleen.



Kuva 4.4. Tangentiaalivirheessä nähtävät yksittäiset poikkeamat.

Näiden tapahtumien satunnaisuudesta johtuen niitä ei pystytä ennustamaan. On siis suo-
tavaa poistaa kyseiset tapaukset opetusdatan joukosta, jotta satunnainen metsä ei pyri
mallintamaan niitä.

Kuvassa 4.5 esitetään virhekäyrää kuvaavan käyrän $ax^2 + bx$ parametrin a käyttäytymistä
ajan funktiona. Voidaan havaita, että parametri värähtelee lähes jatkuvasti vaihtelevalla
amplitudilla. Parametrin a ja b mallintamisen kannalta on kuitenkin järkevämpää pyrkiä
mallintamaan suurimpia piirteitä yksityiskohtien sijaan. Tällöin malli ei kuvaa niin hyvin
käyrän käyttäytymistä, mutta mallin ennustustarkkuus paranee.



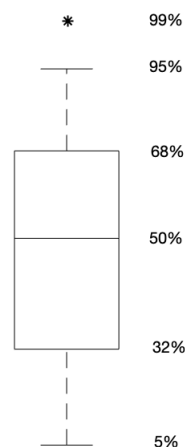
Kuva 4.5. Tangentiaalivirheessä nähtävät yksittäiset poikkeamat.

Kuvan data voitaisiin esimerkiksi alipäästösuodattaa tai käsitellä yksinkertaisella mediaanisuotimella.

5 MALLIN SUORITUSKYKY

Satunnaisen metsän avulla luodaan siis korjausmalli, jolla pystytään ennustamaan normaalin ennustusmallin tuottamien ennustusten virheitä. Malli on opetettu 27 GPS-satelliitilla, joista kustakin on noin 3 vuotta dataa. Muutamia GPS-satelliitteja ei ole käytetty, koska ne ovat olleet liian lyhyen aikaa kiertoradoillaan mallin luomiseksi. Testaukseen tarkoitettua dataa ollaan kerätty noin neljä kuukautta kultakin 27 satelliitilta. Data sisältää 14 päivän ennustuksia, ja niitä tehdään yksi jokaista BE:tä kohti eli kahden tunnin välein. Ennustettuja paikkoja verrataan myöhemmin laskettuihin tarkkoihin sijanteihin ns. *tarkkoihin efemerideihin* (PE, *proadcast ephemeris*). BE:t ja PE:t on ladattu lähteen [4] avoimesta datasta.

Tässä luvussa esitetään tuloksia *laatikkokuvaajien* avulla. Niillä pystytään yksinkertaisesti ilmaisemaan tulosten jakaumat. Tässä diplomityössä määritellään laatikon osat ja *viikset* kuvan 5.1 mukaisesti. Korkein *-merkki kuvaa 99 %-kvantiilia, eli 99 % arvoista esiintyy tämän merkin alapuolella. Ensimmäinen viiksi, eli korkeampi vaakaviiva, ilmaisee, että 95 % arvoista esiintyy viivan alapuolella. Vastaavasti voidaan kuvan mukaisesti päätellä 68 %-, 50 %-, 32 %- ja 5 %-kvantiilit.

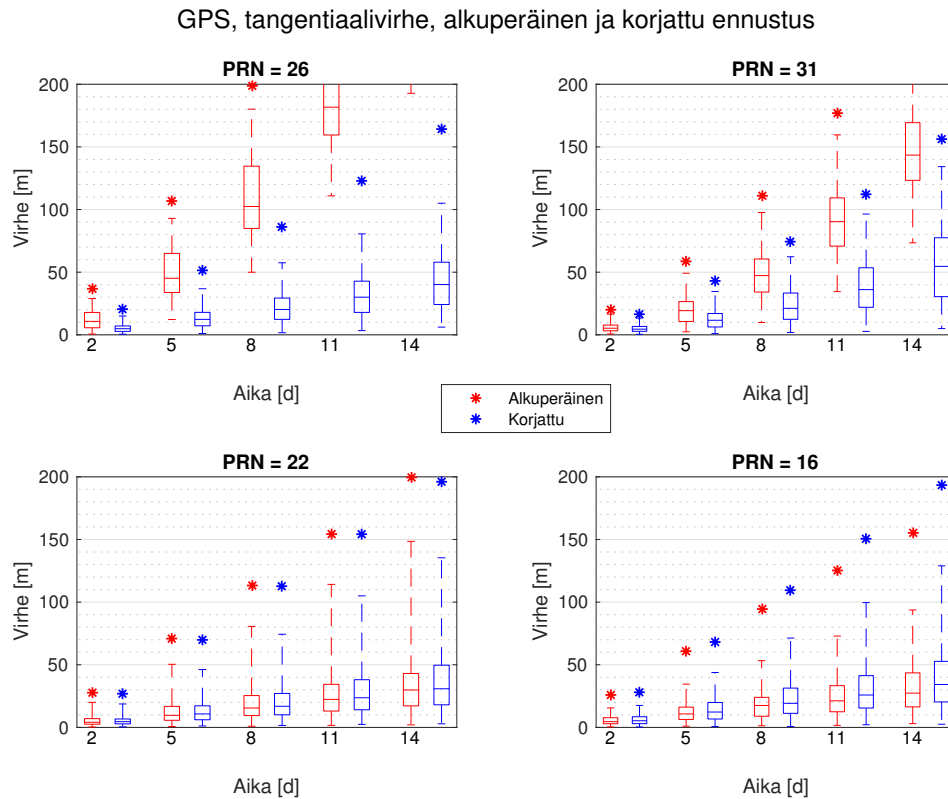


Kuva 5.1. Laatikkokuvaajan tulkinta.

Kaikki virhearvot ovat ilmastuina niiden itseisarvoina. Virhekäyrät ovat todellisuudessa jatkuvia käyriä, mutta poimitaan tasavälein päiviä, joilla tarkastellaan virheen luonnetta. Välin arvot voidaan olettaa lähes lineaarisiksi.

5.1 Parannus virhekomponenteittain

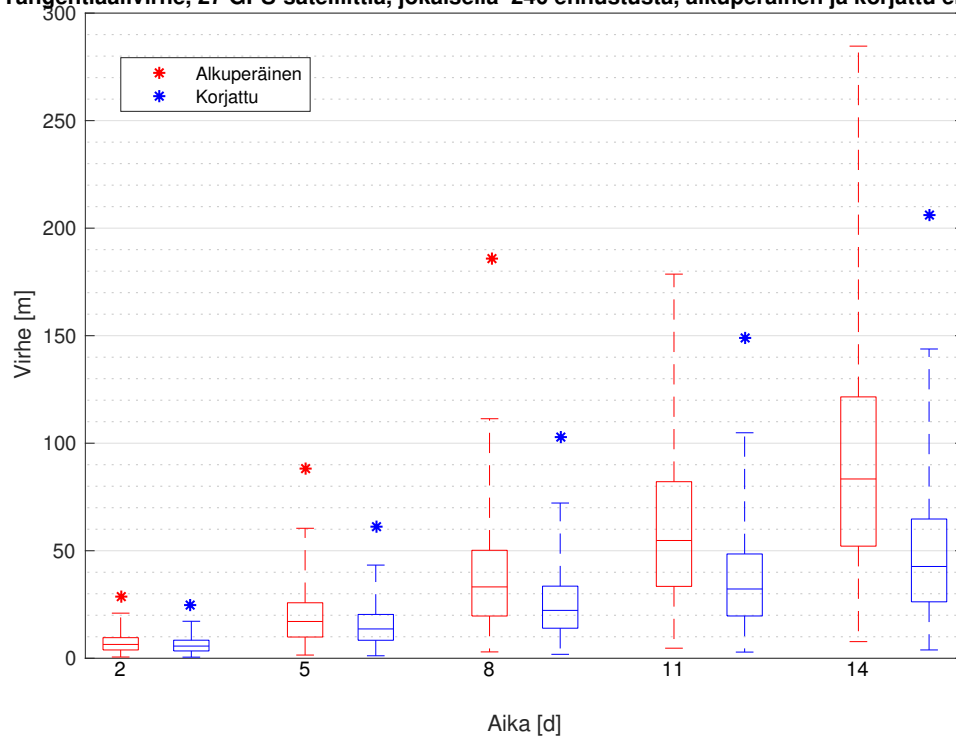
Tangentiaalivirhe oli virhetermeistä suurin, mutta ei kuitenkaan helpoiten ennustettavissa. Kuvaan 5.2 on koottu neljän satelliitin tulokset, jotka on pyritty valitsemaan siten, että ne edustaisivat kaikkia tapaustyyppiejä. Kuvissa on käytetty kerran opetettua mallia. Jakautuma on kerätty 20 päivän ajalta, kahden tunnin välein tehdyiltä ennustuksilta. Jokaisella satelliitilla on siis tehty $20 \cdot 12 = 240$ ennustusta.



Kuva 5.2. Tangentiaalivirheen ennustuksen tulokset, 4 GPS-satelliittia. Punainen kuvaa alkuperäistä ennustusta ja sininen korjattua.

Ensimmäisessä kuvassa ennustusta on onnistuttu parantamaan suuresti. Alkuperäinen virhe eskaloituu yli 200 metriin, mutta korjattuna virheen mediaani on vain noin 40 metriä. Toisessa kuvassa virhe noin puolittuu viimeisillä päivillä. Kolmannessa kuvassa virheen ennustus on vain hieman parantunut. Neljännessä kuvassa korjaus on vain huonontanut alkuperäistä ennustusta, mutta ei suuresti.

Tangentiaalivirhe, 27 GPS-satelliittia, jokaisella 240 ennustusta, alkuperäinen ja korjattu ennustus.

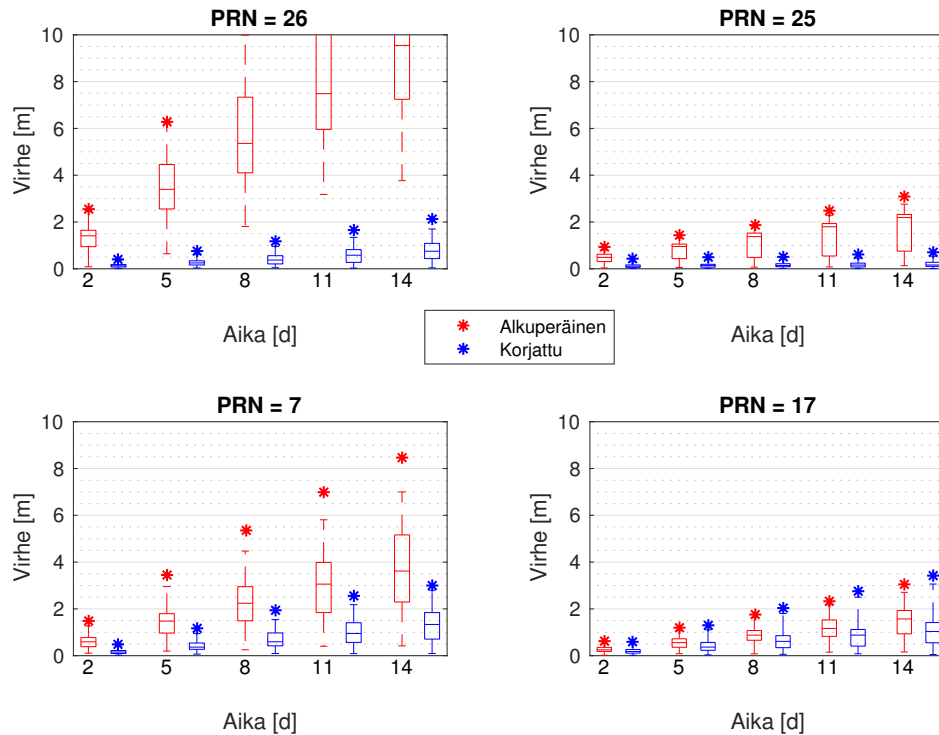


Kuva 5.3. Tangentiaalivirheen ennustuksen tulokset yhteenlaskettuna 27 GPS-satelliitilta. Punainen kuvaa alkuperäistä ennustusta ja sininen korjattua.

Kuvaan 5.3 on kerätty 27 GPS-satelliitin yhteenlaskettu jakauma. Jokaisella satelliitilla on tehty 240 ennustusta, eli yhteensä 6480 ennustusta. Voidaan havaita, että ensimmäisillä päivillä korjaus ei suuresti paranna ennustusta, mutta kahden viikon kohdalla virhe on noin puolittunut kaikkien kvantiilien osalta.

Kuten kaavasta (3.15) nähdään, radiaalivirheen aiheuttama signaalin mittausvirhe on suurin kolmesta virhekomponentista. Satunnaisella metsällä luotiin malli, jolla ennustuksen virhettä saatiin pienennettyä merkitsevästi. Kuvassa 5.4 on esitetty tuloksia neljältä GPS-satelliitilta.

GPS, radiaalivirhe, alkuperäinen ja korjattu ennustus



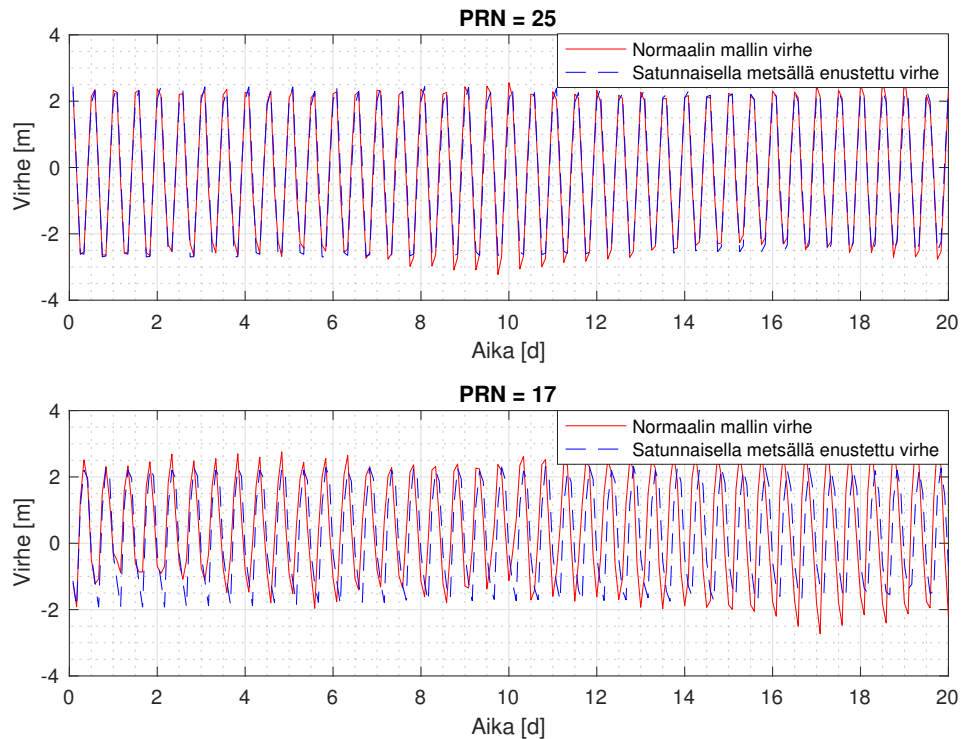
Kuva 5.4. Radiaalivirheen ennustuksen tulokset, neljä GPS-satelliittia. Punainen kuvaa alkuperäistä ennustusta ja sininen korjattua.

Ensimmäisessä kuvassa voidaan havaita suuri parannus normaaliin ennustusmalliin. Toisessa kuvassa radiaalivirhe on saatu kutistettua alle metriin kahden viikon ennustuksessa. Neljännessä kuvassa korjausmalli on hieman huonontanut tuloksia 99 %- ja 95 %-kvantiileilla viimeisillä päivillä. muut kvantiilit ovat kuitenkin parantuneet. Voisi luulla, että satelliitin PRN = 17 korjaus ei toiminut, koska ennustusvirhe oli jo entuudestaan pieni verrattuna muihin satelliitteihin, mutta satelliitilla PRN = 25 korjaus onnistui samoista lähtökohdista.

Kuvassa 5.5 on esitetty näiden kahden satelliitin ennustusvirhettä ja satunnaisella metodilla saadun korjausmallin ulostuloa. Kuva on tehty samalla aikaikkunalla kuin kuva 5.4. Punainen virhekkäyrä kuvaa ennustuksen virhettä 14 päivän kuluttua ennustuksesta. Aika-akselin jokainen kohta kuvaa eri ennustusta. Niitä on tehty 20 päivän ajan jokaisesta BE:stä, eli yhteensä 240 ennustusta. Sininen katkoviiva kuvaa satunnaisen metsän antamaa korjausmallia.

Satelliitilla PRN = 25 huomataan, että korjausmallin antama korjaus toimii hyvin. Molemmat käyrät kulkevat samaa reittiä, jolloin virheen poisto onnistuu niin kuin pitäisi. Satelliitilla PRN = 17 korjaus menee kuitenkin pieleen. Alkuun käyrät kulkevat lähes samaa reittiä, mutta sitten niiden vaiheet eroavat toisistaan. Vaikka korjausmallin antama käyrä on muodoltaan oikea, huonontaa se silti ennustusta.

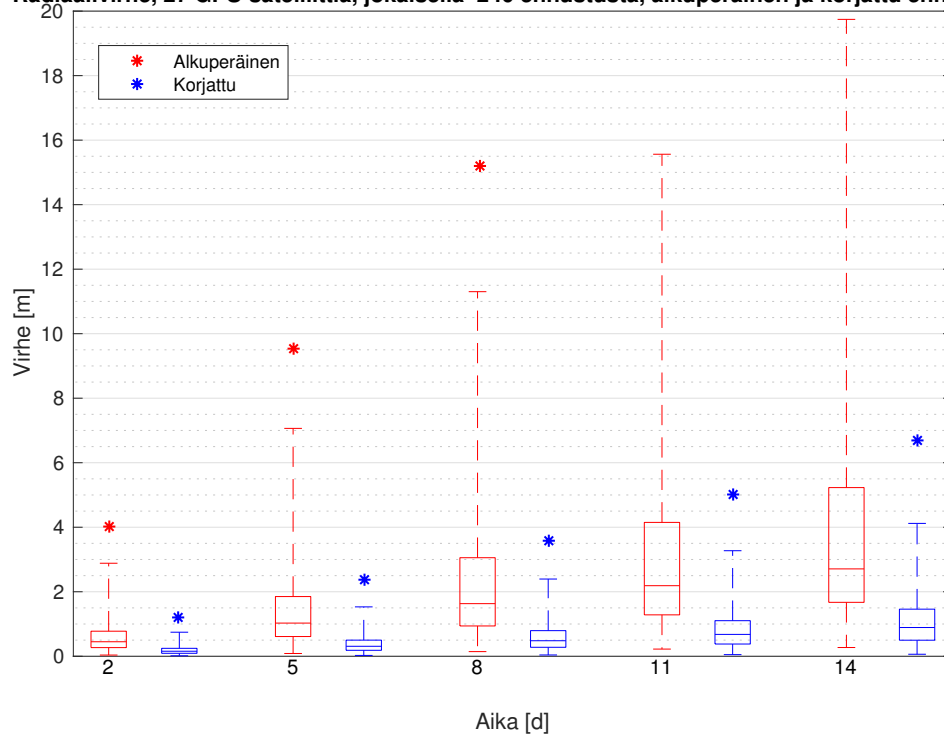
Normaalin ennustusmallin radiaalivirhe 14 päivän kuluttua ja ennustettu virhe



Kuva 5.5. Radiaalivirhe 14 päivän päästä ennustuksesta. Ennustuksia on tehty 20 päivän ajan joka toinen tunti. Sininen katkoviiva kuvaa satunnaisen metsän antamaa arviota ennustusvirheestä.

Kuvassa 5.6 on esitetty mallin toimivuus 27 GPS-satelliitin tulokset yhdistämällä. Voi-
daankin havaita, että mallilla korjatun ennustuksen virhe on vain noin neljäsosan alkupe-
räisestä virheestä.

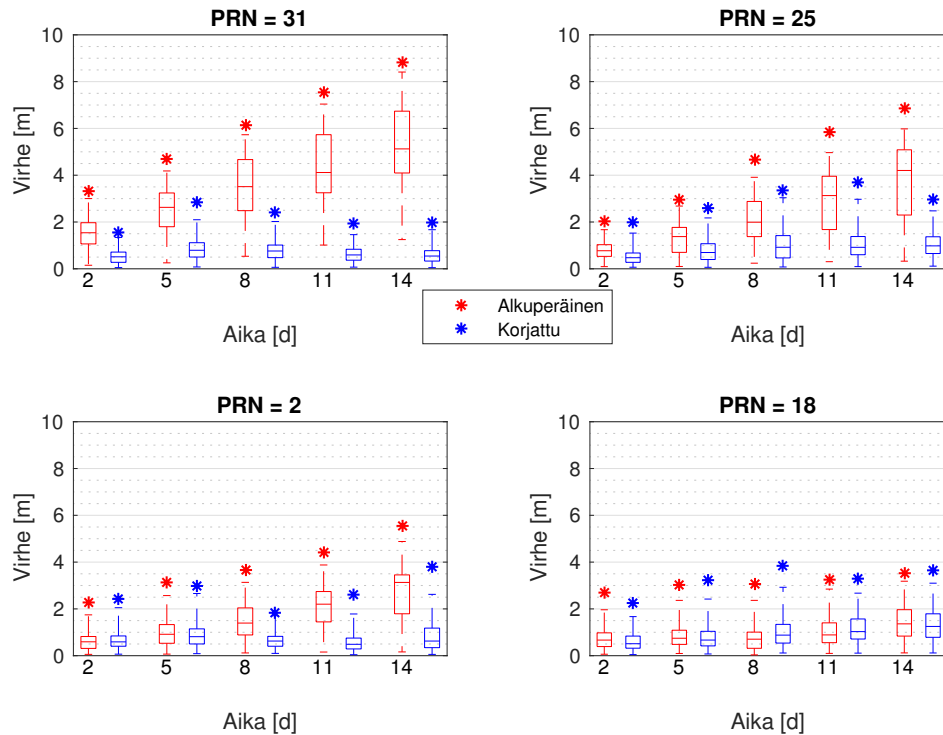
Radiaalivirhe, 27 GPS-satelliittia, jokaisella 240 ennustusta, alkuperäinen ja korjattu ennustus.



Kuva 5.6. Radiaalivirheen ennustuksen tulokset yhteenlaskettuna 27 GPS-satelliitilta. Punainen kuvaa alkuperäistä ennustusta ja sininen korjattua.

Normaalivirheen vaikutus on selvästi radiaalivirhettä merkityksellisempi SISRE:n kanalta (kaava (3.15)), ja sen suuruus on selvästi tangentiaalivirhettä pienempi. Sen vaikutus SISRE:en on selkeästi näistä komponenteista pienin. Kuvaan 5.7 on koottu neljä tapausta eri satelliiteilta. Muiden virhekomponenttien tapaan korjausmalli pystyy parantamaan yksittäisiä ennustuksia paljon ja huonoissa tapauksissa huonontaa ennustusta vain vähän.

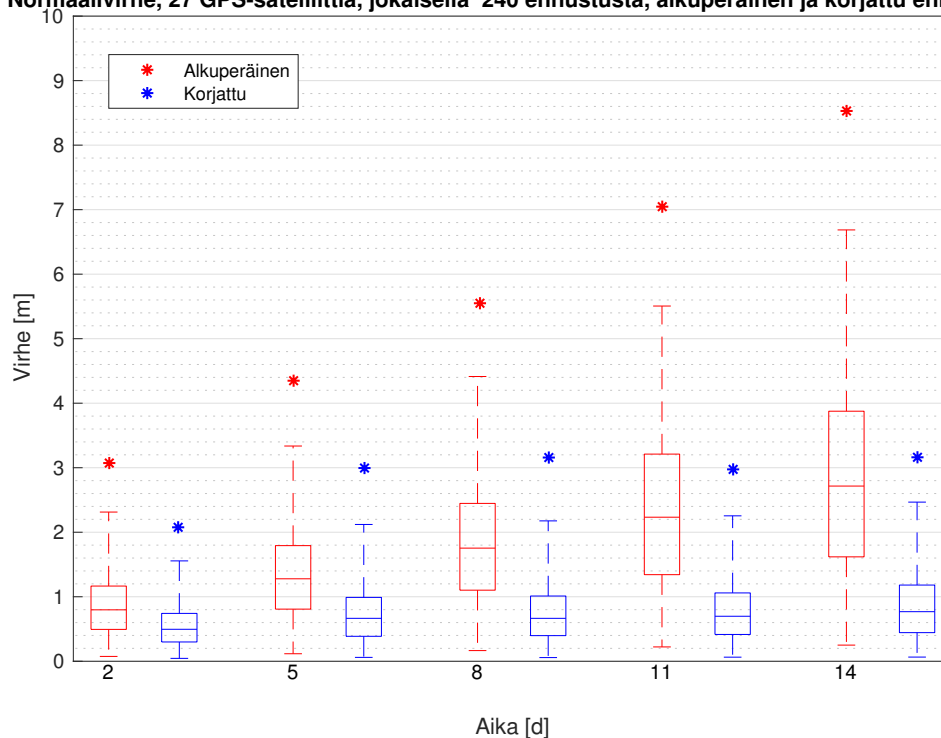
GPS, normaalivirhe, alkuperäinen ja korjattu ennustus



Kuva 5.7. Normaalivirheen ennustuksen tulokset, 4 GPS-satelliittia. Punainen kuvaa alkuperäistä ennustusta ja sininen korjattua.

Kuvassa 5.8 on esitetty kootut tulokset 27 GPS-satelliitista. Vaikka normaali- ja radiaalivirheet ovat käytökseltään samanlaisia, malli ei pysty korjaamaan normaalivirhettä yhtä tehokkaasti kuin radiaalivirhettä. Virhe saadaan pudotettua korjauksella noin kolmasosaan alkuperäisestä kahden viikon kohdalla.

Normaalivirhe, 27 GPS-satelliittia, jokaisella 240 ennustusta, alkuperäinen ja korjattu ennustus.



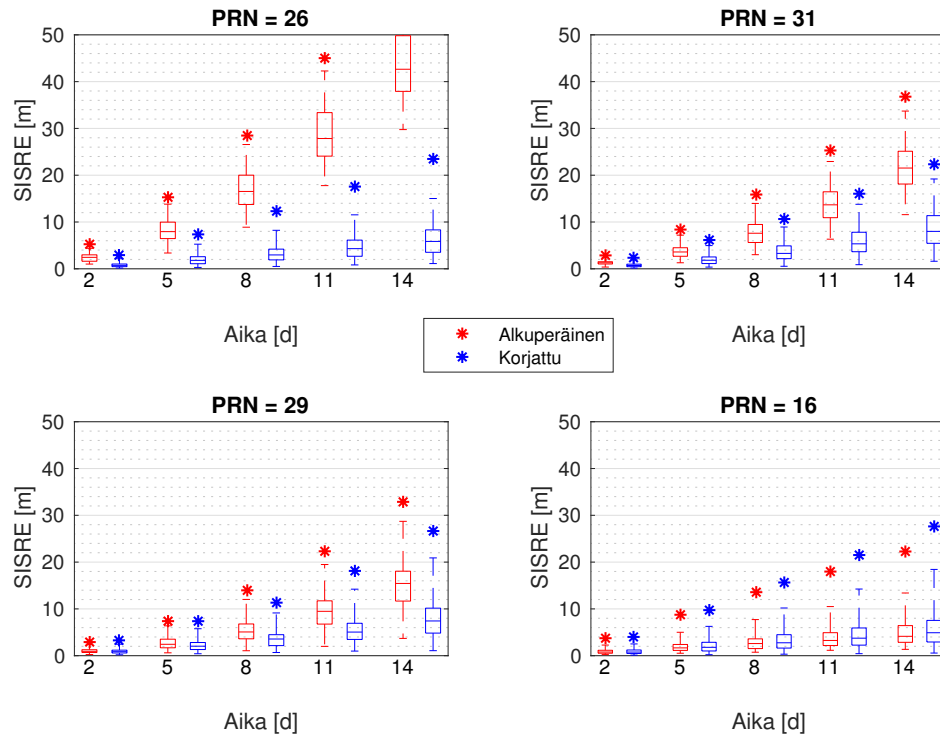
Kuva 5.8. Normaalivirheen ennustuksen tulokset yhteenlaskettuna 27 GPS-satelliitilta. Punainen kuvaa alkuperäistä ennustusta ja sininen korjattua.

Tällä testausjaksolla havaitaan myös, että korjauksella saatu ennustuksen virhe pysyy lähes vakiona 5. päivästä 14. päivään. Kahden viikon kohdalla ennustusvirhe on laskenut alle puoleen alkuperäisestä.

5.2 SISRE

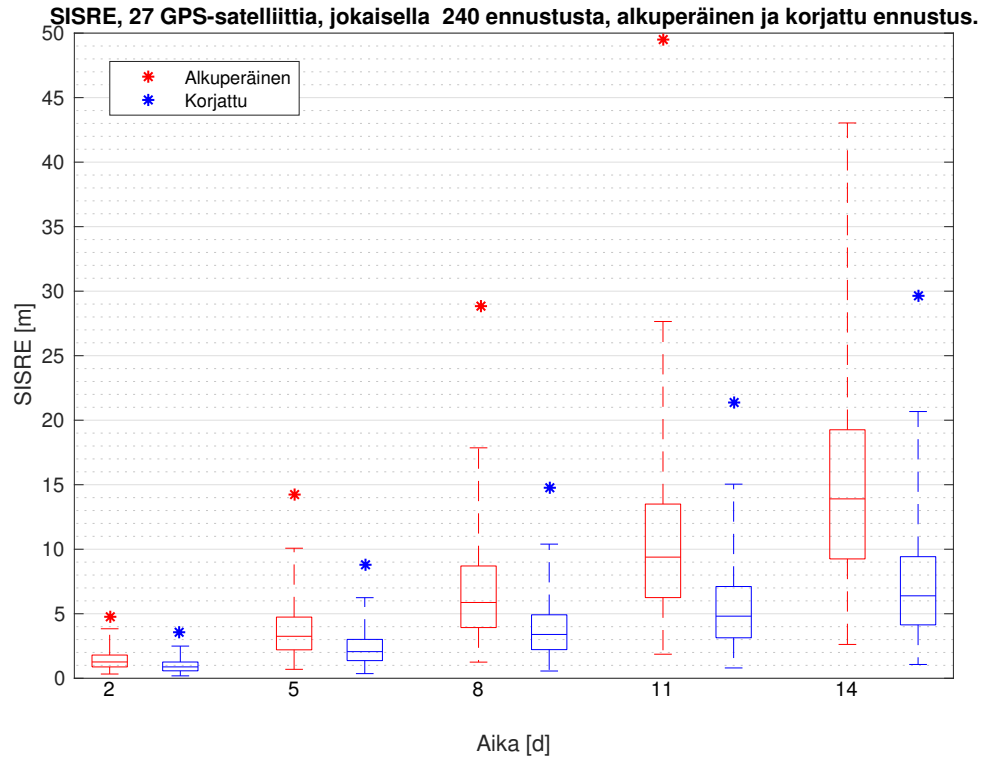
Kuvassa 5.9 on esitetty neljän satelliitin tulokset. Parhaassa tapauksessa malli parantaa ennustusvirhettä huomattavasti, kuten PRN = 26 tapahtui. Kaksi seuraavaa kuvaa edustavat geneeristä tapausta. Ensimmäisinä päivinä malli ei suuresti paranna ennustus-tarkkuutta, mutta muutaman päivän jälkeen mallin toimivuus näkyy. Viimeisessä kuvassa malli huononsi alkuperäistä ennustusta.

GPS, SISRE, alkuperäinen ja korjattu ennustus



Kuva 5.9. Ennustusten SISRE:jen tulokset, 4 GPS-satelliittia. Punainen kuvaa alkuperäistä ennustusta ja sininen korjattua.

Kuvaan 5.10 on koottu tulokset kaikista 27 satelliitista. Tämän testijakson perusteella voidaan päätellä, että korjausmalli toimii hyvin. Kahden viikon ennustuksen jälkeen kaikki virhekvantiilit noin puolittuvat, ja heti 2. päivänä noin yksi kolmasosa virheestä on poistunut.



Kuva 5.10. Ennustusten SISRE:jen tulokset yhteenlaskettuna 27 GPS-satelliitilta. Punainen kuvaa alkuperäistä ennustusta ja sininen korjattua.

Taulukkoon 5.1 on kerätty 27 GPS-satelliitin tulokset. Tulokset on kirjattu viikon ja kahden viikon mittaisilta ennustuksilta. Molemmista tapauksista on kerätty 95 %- ja 68 %-kvantiilit. Satelliiteilla PRN = 11, 16 ja 17 havaitaan, että korjausmalli on aavistuksen huonontanut alkuperäistä ennustusta, mutta yleisesti malli toimii.

GPS, SISRE (m)

PRN	7 päivän ennustus				14 päivän ennustus			
	Q95 %		Q68 %		Q95 %		Q68 %	
	Alkup.	Korj.	Alkup.	Korj.	Alkup.	Korj.	Alkup.	Korj.
1	6.71	5.83	3.22	2.96	18.12	11.04	11.28	5.5
2	9.33	6.17	5.95	3.29	27.85	11.64	20.61	6.57
3	7.8	7.84	4.22	3.85	16.91	15.43	10.51	7.56
5	8.97	6.69	4.76	3.83	25.12	12.34	17.31	6.72
6	26.42	11.07	22.65	5.15	84.31	33.54	74.66	11.97
7	13.36	8.77	9.24	6	36.62	21.22	28.48	14.45
8	10.86	7.32	5.98	3.69	19.04	14.24	10.65	7.13
11	6.79	7.89	3.98	4.07	13.81	19.06	7.42	9.85
12	7.83	6.68	4.32	3.17	21.57	14.72	13.79	7.16
13	11.28	11.2	8.95	5.99	26.3	20.43	18.83	11.63
14	13.16	9.38	10.42	4.41	27.4	17.46	22.66	9.44
15	16.17	12.19	13	6.25	42.31	35.23	30.21	22.42
16	6.85	8.87	3.14	3.88	13.4	18.44	6.43	7.54
17	9.66	12.17	4.95	6.76	23.01	26.64	13.86	14.86
18	12.41	12.07	6.34	5.8	30.73	23.71	16.51	11.16
19	9.13	8.38	5.99	5.59	23.75	20.86	15.83	14.56
20	7.73	8.2	4.2	3.88	21.08	15.8	12.13	7.59
21	9.1	7.03	5.35	3.61	23.62	16.32	15.36	7.44
22	10.08	9.31	3.88	3.52	21.23	19.34	7.49	7.21
23	15.63	10.7	12.04	6.02	31.57	23.14	24.69	11.49
24	9.45	5.76	6.78	3.48	25.33	12.44	19.74	7.32
25	10.67	7.17	4.67	3.1	20.79	13.17	9.71	6.29
26	21.96	7.21	16.32	3.63	61.38	15.02	49.81	8.32
27	12.45	11.41	6.79	4.93	29.13	23.46	14.46	10.61
28	12.03	7.8	7.08	3.75	33.3	16.68	22.75	7.79
29	10.04	8.07	5.56	3.95	28.71	20.9	18.06	10.16
31	11.45	7.5	7.57	3.95	33.71	19.2	25.12	11.36

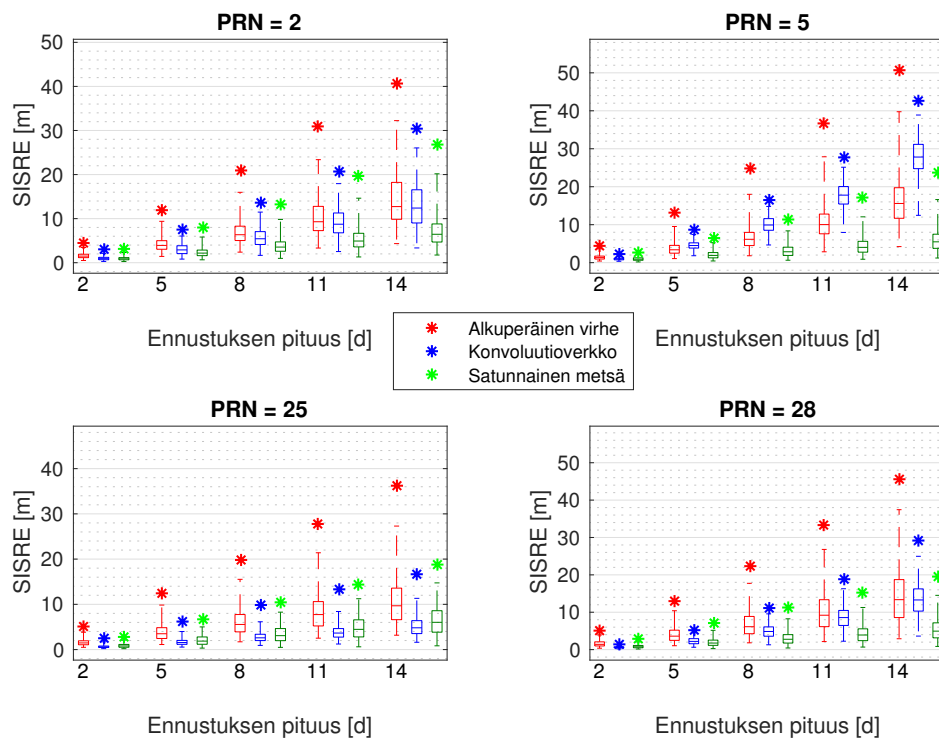
Taulukko 5.1. GPS-satelliiteilla saadut tulokset. SISRE:ä on tarkasteltu 7 ja 14 päivän ennustuksista. Molemmista on kirjattu ylös 95 %- ja 68 %-kvantiilit, joita merkitään sarakkeilla Q95 % ja Q68 %. Alkup.-sarakkeet esittävät normaalin ennustusmallin tuottamat virheet ja Korj.-sarakkeet virheet, kun ennustuksiin on tehty korjausmallin mukaiset korjaukset. Huomaa, että kaikkia GPS-satelliitteja ei ole listattu.

5.3 Vertailu konvoluutioverkolla tuotettuun korjausmalliin

Tässä kappaleessa vertataan satunnaisella metsällä luotua korjausmallia kappaleessa 3.3 mainittuun konvoluutioverkolla luotuun korjausmalliin.

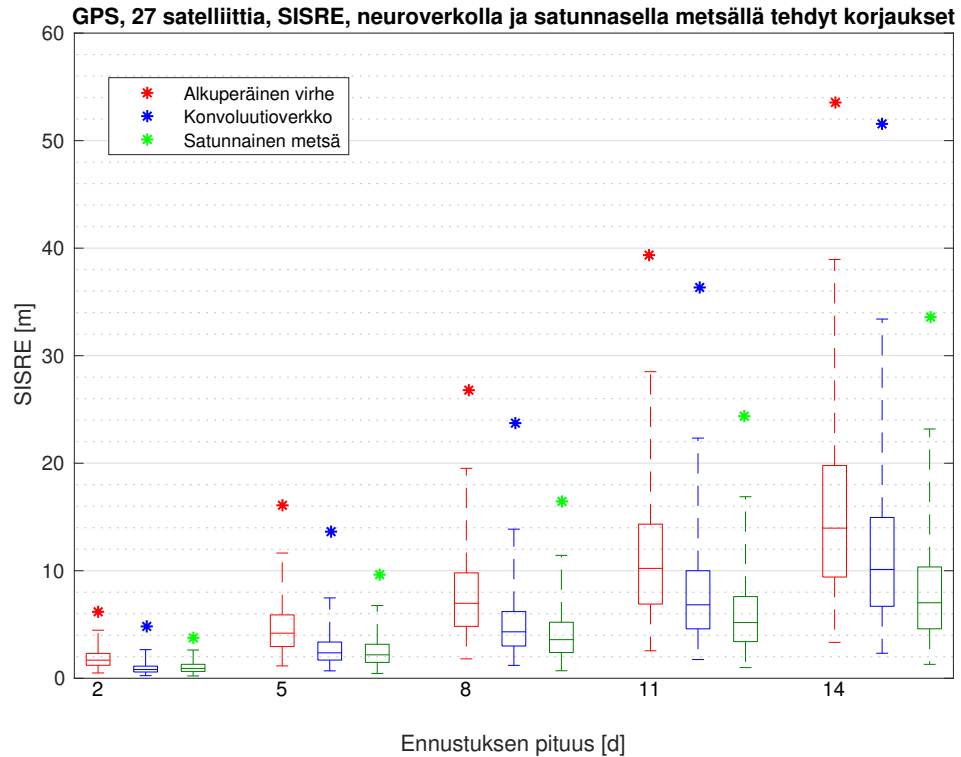
Konvoluutioverkolla tuotettussa mallissa voidaan havaita, milloin tehty korjaus todennäköisesti ei paranna alkuperäistä ennustusta. Kun mahdollinen huono korjaus tunnistetaan, käytetään alkuperäistä ennustusta sellaisenaan ilman korjausta. Näitä tapauksia on mukana käytetyssä datassa.

GPS, SISRE, neuroverkolla ja satunnasella metsällä tehdyt korjaukset



Kuva 5.11. SISRE, vertailussa alkuperäinen virhe sekä konvoluutioverkolla ja satunnaisella metsällä parannellut ennustukset. Siniset laatikot kuvaavat konvoluutioverkolla korjattuja ennustuksia ja vihreät laatikot satunnaisella metsällä korjattuja. Punaiset laatikot esittävät alkuperäisen ennustuksen virhettä.

Kuvassa 5.11 on esitetty tulokset neljältä GPS-satelliitiltä. Ensimmäinen kuva edustaa tyypillistä tapausta. Konvoluutiolla paranneltu ennustus parantaa yleensä normaalia ennustusmallia, mutta satunnaisella metsällä korjattu ennustus tuottaa yleisesti paremman ennustuksen. Kaikissa tapauksissa ei kuitenkaan ole näin, esimerkiksi satelliitilla PRN = 25 konvoluutioverkolla tehty korjausmalli toimi satunnaisella metsällä tehtyä mallia paremmin.



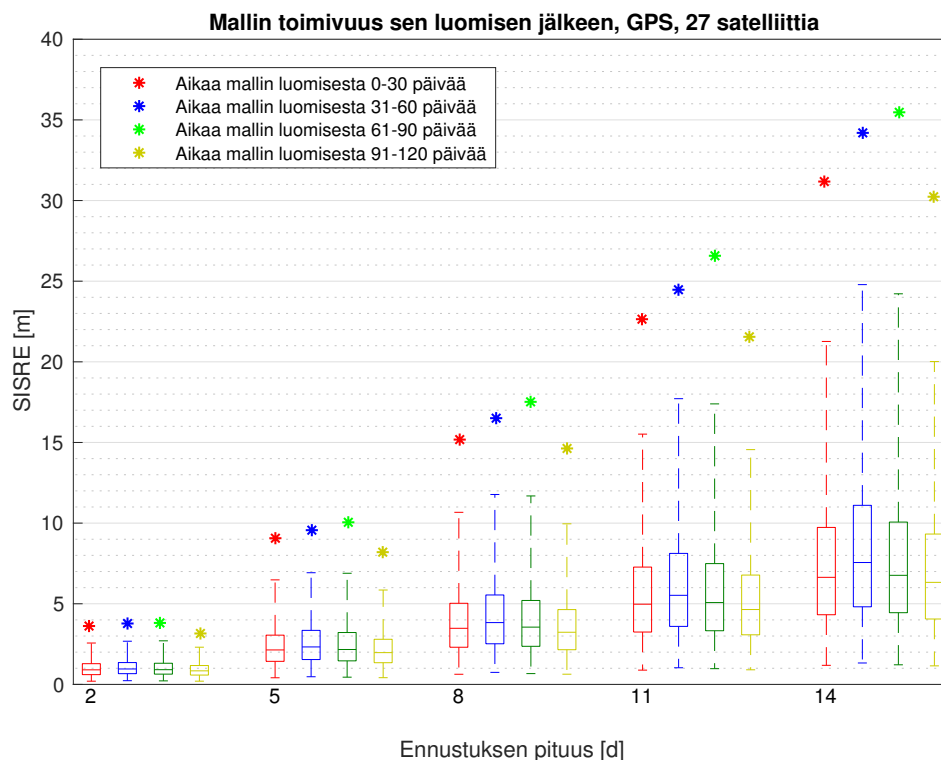
Kuva 5.12. SISRE, 27 satelliittia. Vertailussa alkuperäinen virhe, konvoluutioverkolla ja satunnaisella metsällä paranneltu ennustus.

Kuvassa 5.12 on esitetty yhteenkootut tulokset 27 GPS-satelliitilta. Kuten kuvasta huomaa, satunnaisella metsällä tuotettu malli tuottaa tällä testijaksolla paremmat korjaukset rataennustuksiin kuin konvoluutioverkko. Lyhyillä ennustuksilla konvoluutioverkko voi tuottaa yhtä hyvän tai jopa paremman korjauksen, mutta pitkillä ennustusajoilla satunnainen metsä kykenee parempiin tuloksiin. Konvoluutioverkon onnistuminen lyhyillä ennustuspätkillä selittyy sillä, että silloin virheellä on selkeä riippuvuus taaksepäin tehtyyn ennustukseen. Satunnaisella metsällä luodun mallin etu lienee se, että aika ei ole yksi konvoluutioverkon syöteparametreista, sillä tässä diplomityössä on havaittu suuri riippuvuus ajan ja rataennustuksen virheen välillä.

5.4 Parametrien luonti etukäteen

Yksi tämän diplomityön tutkimuksen periaatteista on, että paikannuslaite ei pysty vastaanottamaan uusia avusteparametreja jatkuvasti, vaan esimerkiksi vain WLAN-yhteydessä. Tällöin etu on, että parametrit olisivat tarpeeksi pitkäikäisiä, tai että niitä pystytäisiin lähettämään laitteelle etukäteen. Yksittäisten parametrien ikä ei korjausmallilla ole kovin pitkä, joten ideaalista olisi pystyä tuottamaan parametreja tulevaisuuteen tarpeeksi ajoissa.

Kuvassa 5.13 on esitetty, kuinka hyvin korjausmalli suoriutuu parametrien luomisesta tulevaisuuteen. 99 %-kvantiililla näyttäisi olevan lievä kasvava trendi mallin vanhetessa, mutta muilla kvantiileilla vastaavaa ei ole havaittavissa. Voidaan siis olettaa, että parametrien vanhenemisella ei ole vahvaa riippuvuutta mallin suorituskykyyn. Yhdellä mallilla voidaan siis ennustaa korjausparametreja kuukausien päähän, ja ladata niitä valmiiksi paikannuslaitteelle, kun se on mahdollista.

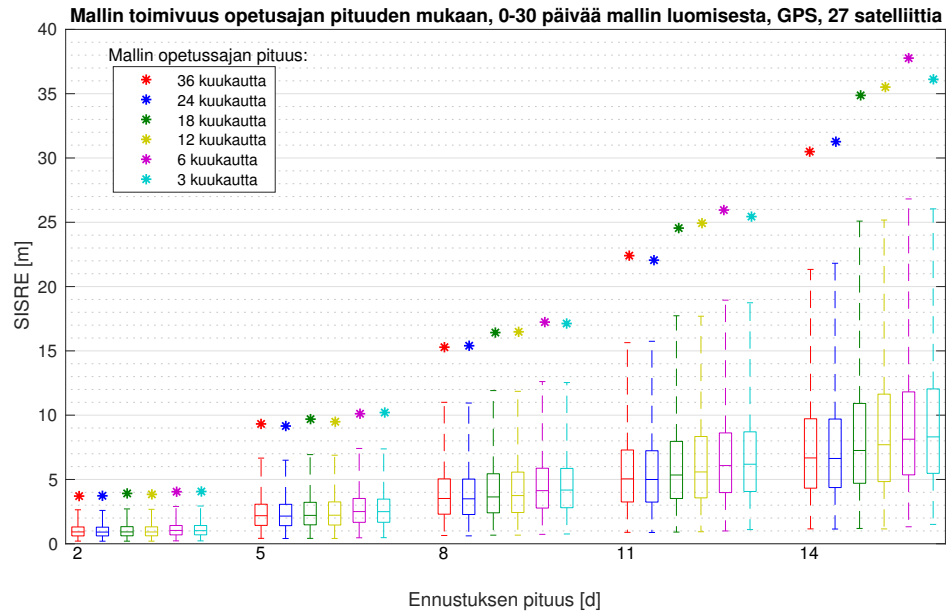


Kuva 5.13. SISRE, 27 satelliitilta, kun korjausmallin luomisesta on kulunut vastaavat ajat.

5.5 Opetusdatan koko

Aikaisemmissa tuloksissa on käytetty korjausmallia, jonka luomiseen on käytetty noin kolmen vuoden mittaisen ajanjakson dataa GPS-satelliittien rataennustuksista. Uusia paikannussatelliitteja lisätään kuitenkin kiertoradalle aika ajoin lisää, sekä vanhoja satelliitteja korvataan uusilla. Täten olisi suotavaa, että kun uusi satelliitti otetaan käyttöön, pystyttäisiin mahdollisimman lyhyessä ajassa luomaan sille korjausmalli.

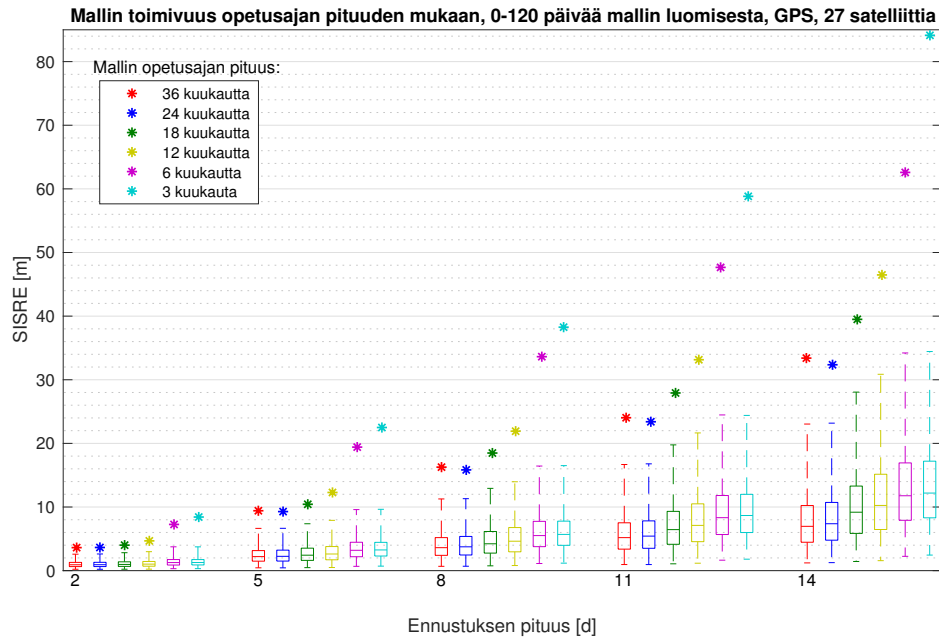
Kuvassa 5.14 on esitetty, miten mallin tekemiseen käytetty datan määrä vaikuttaa mallin tarkkuuteen. Pienillä ennustuspituuksilla ei havaita suuria eroja, mutta ennustuksen loppupäässä on havaittavissa trendi, jossa lyhyempi opetus aika tuottaa epätarkemman mallin. Kahden ja kolmen vuoden opetusajoilla tuotettujen mallien toimivuudessa ei ole juurikaan eroja.



Kuva 5.14. Korjausmallin luomiseen käytetyn datan määrän vaikutus mallin toimivuuteen. Testauksen ajanjaksona on käytetty seuraavaa kuukautta mallin luomisesta.

Alle vuoden mittaisella opetusajalla tuotetut korjausmallit eivät luonnollisesti opi koko vuoden mittaista periodista käyttäytymistä. Nämä mallit kuvaavat enemmän satelliittien lokaalia mallia tietyssä vuodenaikaympäristössä. Vastaavasti tasan vuoden mittaisella opetusajalla tuotettu malli ei mallinna optimaalisesti vuoden periodista vaihtelua, koska ei ole nähnyt periodisuuden tapahtuvan uudestaan seuraavana vuonna.

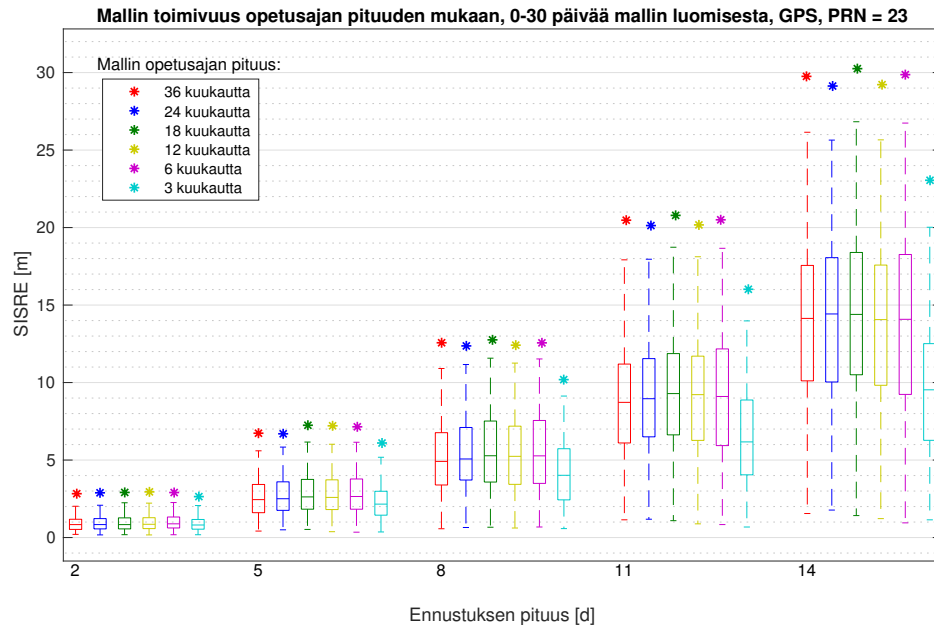
Kuvassa 5.15 on esitetty samojen mallien toimivuus, kun niitä käytetään pidempi ajanjakso. Tällöin lokaalit korjausmallit eivät enää toimi niin hyvin, joten erot opetusajakojen pituudella kasvavat. Varsinkin 99 %-kvantiililla kolmen ja kuuden kuukauden ajalla tehdyt mallit erottuvat selvästi muista malleista. Taaskaan kahden ja kolmen vuoden opetusajoilla ei havaita suurta eroa. Kahden vuoden opetusajaksolla tuotettu malli näyttää jopa toimivan hieman paremmin tällä testidatalla. Tämä saattaa johtua vain kohinasta, ja vakuuttavampien tulosten saamiseksi tätä tarvitsisi tutkia enemmän.



Kuva 5.15. Korjausmallin luomiseen käytetyn datan määrän vaikutus mallin toimivuuteen. Testauksen ajanjaksona on käytetty seuraavaa 4 kuukautta mallin luomisesta.

Mielenkiintoisena poikkeuksena huomautettakoon, että kolmen kuukauden opetusajalla tuotettu korjausmalli toimii lyhyellä testausjaksolla paremmin kuin kuuden kuukauden opetusajalla tehty malli (kuva 5.14). Tämä johtunee siitä, että opetusajan ollessa lyhyt korjausmallista tulee hyvin lokaali malli, joka toimii opetusajanjakson läheisyydessä. Testiajanjakson pidentyessä kolmen kuukauden opetusajan heikkous tulee ilmi. Kuvasta 5.15 nähdään, kuinka 99 %-kvantiili kasvaa selkeästi opetusajan lyhentyessä.

Kuvassa 5.16 on esitetty GPS-satelliitin PRN = 23 vastaava kuva kuukauden testijaksolla. Tällä satelliitilla kolmen kuukauden opetusdatalla tehty lokaali korjausmalli tuottaa tarkempia tuloksia muihin korjausmalleihin verrattuna. Satelliitin ennustusvirhe siis käytäytyy opetuksen ja testauksen aikana hyvin samankaltaisesti. Muutamilla muillakin satelliiteilla on saatu samankaltaisia tuloksia. Yleinen trendi on kuitenkin kuvan 5.14 mukainen. Kun korjausmallia käytetään pidemmässä ajanjaksossa, lokaalin mallin tehokkuus katoaa, ja globaalit mallit tuottavat tarkempia tuloksia.



Kuva 5.16. Korjausmallin luomiseen käytetyn datan määrän vaikutus mallin toimivuuteen GPS-satelliitilla PRN = 23. Testauksen ajanjaksona on käytetty seuraavaa kuukautta mallin luomisesta.

Satelliiteista tulisi siis kerätä dataa vähintään kaksi vuotta, jotta saataisiin tarpeeksi dataa optimaalisen globaalimallin luomiseen. Lokaalit mallit saattavat toimia joillakin satelliiteilla paremmin lyhyen aikaa, mutta yleisesti tarkempiin tuloksiin päästään globaaleilla malleilla. Vertailemalla kuvia 5.14 ja 5.10 havaitaan kuitenkin, että lyhyen aikaa opetettu lokaali korjausmalli parantaa normaalia ennustusmallia. On myös huomioitava, että jos satelliittien ennustukseen käytettyä ennustusmallia muutetaan esimerkiksi SRP-parametrien (*Solar Radiation Pressure*) suhteen [11], uutta 2 vuoden opetusdataa ei tarvitse odottaa, vaan aikaisempia ennustuksia voidaan simuloida päivitetyllä ennustusmallilla.

6 YHTEENVETO

Satunnaisen metsän avulla pystytään luomaan korjausmalli, joka parantaa huomattavasti paikannussatelliittien rataennustuksia. Korjausmalli ennustaa normaalin ennustusmallin virhettä satelliitin PRN-numeron ja ajan funktiona. Mallin ulostulona on satelliitin rataennustuksen RTN-virhekomponentit 14 päivän mittaiselta ennustukselta. Korjausmallin antama virhe vähennetään rataennustuksista, ja saadaan tarkemmat ennustukset satelliittien radoista.

Ennustetun radan SISRE pienenee noin puoleen alkuperäisestä ennustuksesta (kuva 5.10) kahden viikon mittaisilla ennustuksilla. Parametreja pystytään luomaan samalla mallilla myös kuukausia tulevaisuuteen, joten korjausmallia ei ole täytynyt opettaa tuoreimmalla datalla. Testidatalla 3–4 kuukauden päähän tehdyt korjausparametrit toimivat yhtä hyvin kuin 1–30 päivän päähän tehdyt ennustukset mallin luomisesta (kuva 5.13).

Ennustusvirhettä pystyttäisiin mallintamaan myös pidemmille ennustuksille, esimerkiksi 30 päivän mittaisille virheille, mutta tällöin satelliitin kellon virhe on kasvanut jo liian suureksi paikannustarkoitukseen, eikä rataennustuksen parantaminen ole enää mielekäästä.

Satunnaisella metsällä tuotettujen mallien etuna aikaisemmin kokeiltuihin metodeihin on hyvä ennustustarkkuus sekä se, että yhdellä mallilla pystytään ennustamaan kuukausien päähän. Tällöin paikannuslaite ei tarvitse jatkuvaa internetyhteyttä korjausten tekemiseksi. Huonona puolena on vastaavasti suuri datamäärä opetusta varten, sillä satelliitin on täytynyt olla radallaan noin kaksi vuotta optimaalisten tulosten saamiseksi.

Tässä diplomityössä kaikki tulokset on saatu GPS-satelliittien ennustuksista. GNSS-konstellatiot ovat kuitenkin hyvin samankaltaisia keskenään, ja on oletettavaa, että samalla periaatteella voidaan luoda satunnaisen metsän avulla korjausmallit myös muille konstellaatioille.

LÄHDELUETTELO

- [1] G. Biau. Analysis of a Random Forests Model. *The Journal of Machine Learning Research* 13 (2012), 1063–1095.
- [2] L. Breiman. Random Forests. *Machine Learning* 45.1 (2001), 5–32.
- [3] L. Breiman. *Manual on setting up, using, and understanding random forests v3.1*. Statistics Department University of California Berkeley, CA, USA, 2002.
- [4] *CDDIS - NASA's archive of Space Geodetic Data*. 28. tammikuuta 2018. URL: https://cddis.nasa.gov/Data_and_Derived_Products/GNSS/GNSS_data_and_product_archive.html (viitattu 28.01.2018).
- [5] D. Gebre-Egziabher ja S. Gleason. *GNSS Applications and Methods*. Artech House, Boston, 2009.
- [6] R. Genuer, J. Poggi ja C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters* 31.14 (2010), 2225–2236.
- [7] P. Geurts, D. Ernst ja L. Wehenkel. Extremely randomized trees. *Machine Learning* 63.1 (2006), 3–42.
- [8] *Global Positioning Systems Directorate Systems Engineering Integration Interface Specification IS-GPS-200*. Versio J. 2018. URL: <https://www.gps.gov/technical/icwg/IS-GPS-200J.pdf>.
- [9] P. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. 2nd edition. Artech House, Boston, 2013.
- [10] E. Kaplan ja C. Hegarty. *Understanding GPS/GNSS : Principles and Applications*. 3rd edition. Artech House, Boston, 2017.
- [11] H. Kosola. GNSS-satelliitin ennustetun kiertoradan esittäminen rataparametreinä. *Tampereen teknillinen yliopisto* (2012).
- [12] G. Louppe, L. Wehenkel, A. Suter ja P. Geurts. Understanding variable importances in forests of randomized trees. *Advances in neural information processing systems* 26 (2013), 431–439.
- [13] K. P. Murphy. *Machine Learning, A Probabilistic Perspective*. The MIT Press, 2012, s. 479-551.
- [14] *NGA. EGM2008 coefficients*. 28. tammikuuta 2018. URL: http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/first_%20release.html (viitattu 28.01.2018).
- [15] H. Peng ja X. Bai. Improving Orbit Prediction Accuracy through Supervised Machine Learning. *Advances in Space Research* 61.1 (2018).
- [16] J. Pihlajasalo. Improvement of satellite orbit prediction accuracy and quality with deep learning and spectral analysis. *Tampereen teknillinen yliopisto* (2017).
- [17] S. Pohjalainen ja L. Paunonen. *Matriisilaskenta MAT-60000*. TTY Matematiikan laitos, 2015, s. 44-50.

- [18] M. Poutanen. *Satelliittipaikannus*. Tähtitieteellinen yhdistys Ursa ry, 2016.
- [19] A. Pukkila. GNSS satelliitin radan ennustamisen virhetermien analysointi. *Tampereen teknillinen yliopisto* (2014).
- [20] M. Seppänen. GPS-satelliitin radan ennustaminen. *Tampereen teknillinen yliopisto* (2010).
- [21] K. Väisälä. *Geometria*. WSOY, Porvoo, 1961.